

181161-2

by Pitawat Koonkamoot

Submission date: 18-Nov-2018 09:37PM (UTC+0700)

Submission ID: 1041185835

File name: Report181161-2.pdf (2.66M)

Word count: 12905

Character count: 38658

ห้องสอบ 5

CE61-12

ถุงมือสำหรับตรวจจับท่าทางมือ

SMART GLOVE FOR GESTURE RECOGNITION

พัทธิวิรชุณภูวาน

พิทวัส คุณกะมุต

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ถุงมือสำหรับตรวจจับท่าทางมือ

นายพัทธิวีร์ ชุมกร
นายพิทวัส คุณกานนท
พศ. ชนา วงศ์สุวรรณ
อาจารย์ที่ปรึกษา
ปีการศึกษา 2561

บทคัดย่อ

เทคโนโลยีถุงมือตรวจจับท่าทางมือในปัจจุบันอาจถูกพัฒนามาเพื่อวัตถุประสงค์ใดอย่างหนึ่ง เช่น เพื่อใช้งานร่วมกับเทคโนโลยี Virtual Reality และเพื่อใช้แปลภาษามือเป็นคำหรือตัวสะกดภาษาอังกฤษ แต่สำหรับท่าทางที่แสดงถึงภาษาไทยนั้นจะมีท่าที่แตกต่างออกไปโดยเฉลยทางตัวสะกดไทยที่ไม่สามารถนำถุงมือที่แปลภาษามือที่เป็นภาษาอังกฤษมาใช้ได้โดยตรง เพราะเกิดข้อจำกัดเรื่องท่าทางที่สามารถตรวจจับได้ในด้านฮาร์ดแวร์หรือซอฟต์แวร์

โครงการนี้จึงมีเป้าหมายเพื่อพัฒนาถุงมือขึ้นมาใช้ร่วมกับเทคโนโลยี Machine Learning ที่ได้รับการพัฒนาประสาทเชิงพื้นและมีความนิยมในปัจจุบันเพื่อตรวจจับท่าทางมือ โดยเริ่มจากท่าทางที่แสดงถึงตัวสะกดไทย และเพื่อลดข้อจำกัดเรื่องท่าทางสำหรับการใช้งานอื่น ๆ ในอนาคตจะมีแอปพลิเคชันให้ผู้ใช้งานรถตั้งค่าชุดของท่าทางที่จะใช้เป็นอินพุตและการกระทำที่ต้องการจากท่าทางที่ตั้งค่าไว้ได้ ตัวอย่างเช่น การแสดงผลรูปแบบแอปพลิเคชัน แสดงข้อมูลบนหน้าจอ HTTP ไปยังอุปกรณ์อื่นในเครือข่ายเป็นต้น ดังนั้นถุงมือจะไม่เพียงแค่สามารถตรวจจับท่าที่แสดงถึงตัวสะกดไทยเท่านั้น แต่จะสามารถตรวจจับท่าทางที่เป็นการเคลื่อนไหวได้หลากหลายแบบ หลากหลายท่าติดต่อ กันได้ และสามารถทำได้มากกว่าเพียงแค่แสดงผลเพื่อการใช้งานที่หลากหลายวัตถุประสงค์มากขึ้น

ถุงมือที่ใช้ในการพัฒนาประกอบด้วยหน่วยไมโครคอนโทรลเลอร์ NanoPI-DUO เพื่อใช้ประมวลผล และเซนเซอร์หน่วยวัดอัตราหมุนเวียนแบบ Pololu MinIMU9-v5 6 ตัวเพื่อวัดมุมของหลังมือและนิ้วห้า 5 นำมารวบรวมเป็นมุมด้วยขั้นตอนของตัวกรองร่วมสองแบบ และประมวลผลเป็นท่าทางด้วยโมเดล โครงข่ายประสาทเทียมแบบ LSTM โดยขอบเขตของถุงมือคือสามารถแยกท่าทางที่เป็นภาษาอังกฤษตัวสะกดไทยได้อย่างน้อย 30 ตัว และสามารถตั้งค่าท่าทางดังกล่าวเพื่อแสดงผลเป็นค่าอื่นได้ผ่านเว็บแอปพลิเคชัน

สารบัญ

	หน้า
4 บทคัดย่อภาษาไทย	I
กิตติกรรมประกาศ.....	II
สารบัญ	III
สารบัญตาราง	IV
สารบัญรูป	V
 บทที่ 1 บทนำ	 1
1.1 ความเป็นมาของมูลหา.....	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตของโครงการ	3
1.5 ข้อจำกัดของโครงการ	3
1.6 แผนการดำเนินงาน.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1 เทคโนโลยีด้านสารดิจิทัล	6
2.2 เทคโนโลยีด้านการประมวลผลท่าทางมือ	10
2.3 เทคโนโลยีด้านฐานข้อมูล	15
2.4 เทคโนโลยีด้านแอพพลิเคชัน.....	15
2.5 ทฤษฎีเกี่ยวกับการคำนวณ	17
2.6 งานวิจัยที่เกี่ยวข้อง.....	20
บทที่ 3 การออกแบบและพัฒนา.....	24
3.1 การออกแบบระบบโดยรวม	24
3.2 การออกแบบถุงมือ	25
3.3 การออกแบบเว็บแอพพลิเคชัน	28
3.4 การออกแบบฐานข้อมูล.....	31
3.5 การออกแบบชุดข้อมูล.....	31
3.6 การออกแบบโมเดล Machine Learning	32

บทที่ 4 การทดลอง	36
4.1 การทดลองการเก็บข้อมูลทั้งสองแบบ	38
4.2 การทดลองการปรับปรุงและประยุกต์ค่าจากกลุ่มมือ.....	39
4.3 การทดลองเกี่ยวกับเว็บแอพพลิเคชัน	42
4.4 การทดลองเกี่ยวกับ Machine Learning	44

สารบัญตาราง

	หน้า
1.1 แผนการดำเนินงาน	4
2.1 ข้อมูลจำเพาะ Nanopi-DUO	6
2.2 ข้อมูลจำเพาะเซนเซอร์ Polulu MinIMU9-v5	8
4.1 ผลการทดสอบความแม่นยำโหมดเดลเบนที่ 1	45
4.2 Confusion Matrix ของโหมดเดลเบนที่ 1	45
4.3 ผลการทดสอบความแม่นยำโหมดเดลเบนที่ 2	46
4.4 Confusion Matrix ของโหมดเดลเบนที่ 2	46
4.5 ผลการทดสอบความแม่นยำโหมดเดลเบนที่ 3	48
4.6 Confusion Matrix ของโหมดเดลเบนที่ 3	48

สารบัญ

หัวข้อ	หน้า
1. คุณมีอุปกรณ์ที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality	1
2. คุณมีอุปกรณ์ที่ใช้เปลกภาษาไม่อื่น	1
2.1 แผนภาพโครงสร้าง, ขาเข้า/ขาออกของ NanoPI-DUO	7
2.2 Polulu MinIMU9-v5	8
2.3 SSD1306	9
2.4 ตัวอย่างเครื่องข่ายการเรียนต่อของบัส I2C	9
2.5 TCA9548A	10
2.6 สัญลักษณ์ TensorFlow	10
2.7 สถาปัตยกรรมโครงข่ายประสาทเทียม	11
2.8 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ RNN	12
2.9 สถาปัตยกรรมโครงข่ายประสาทเทียมแบบ LSTM	15
2.10 สัญลักษณ์ MongoDB	15
2.11 สัญลักษณ์ VueJS	16
2.12 สัญลักษณ์ Flask	16
2.13 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูลระหว่าง REST กับ WebSocket	17
2.14 ระบบของตัวกรองร่วมสองแบบ	18
2.15 ระบบของตัวกรองค่ามาตรา	19
2.16 คุณมีของงานวิจัย A Gesture Detection Glove For Human-computer Interaction	20
2.17 Rotation Matrix	21
2.18 แผนภูมิเปรียบเทียบความแม่นยำและเวลาที่ใช้ระหว่างโมเดล Machine Learning แต่ละตัวของ งานวิจัย A Gesture Detection Glove For Human-computer Interaction	21
2.19 แผนภูมิเวลาในการวิเคราะห์ทำของงานวิจัย A real-time continuous gesture recognition system for sign language	22
2.20 คุณมีจากงานวิจัย 3-D hand motion tracking and gesture recognition using a data glove	23
3. โครงสร้างระบบ	25
3.2 ขั้นตอนการทำงานแบ่งตามส่วนของการตรวจท่าทาง	25
3.3 โครงสร้างคุณมือแบบที่ 1	26

3.4 โครงสร้างคุณมือแบบที่ 2	27
3.5 แผนผังลำดับการทำงานของการตรวจจับท่าทาง	28
3.6 ต้นแบบหน้าตาเว็บแอพพลิเคชั่นส่วนของผู้ใช้	29
3.7 แผนผังลำดับการทำงานของการเรียกคู/เพิม/แก้ไข/ลบ ท่าทางของผู้ใช้	29
3.8 ต้นแบบหน้าตาเว็บแอพพลิเคชั่นส่วนของนักพัฒนา	30
3.9 แผนผังลำดับการทำงานของการบันทึกและส่งออกชุดข้อมูลของนักพัฒนา	30
3.10 แผนภาพฐานข้อมูล	31
3.11 สถาปัตยกรรมโครงข่ายประชาทีเยมโ้มเดลที่ 1	33
3.12 สถาปัตยกรรมโครงข่ายประชาทีเยมโ้มเดลที่ 2	34
3.13 สถาปัตยกรรมโครงข่ายประชาทีเยมโ้มเดลที่ 3	35
4.1 ทำนิ่งที่ใช้ในการทดสอบ	36
4.2 ทำเคลื่อนไหวที่ใช้ในการทดสอบ - พยัญชนะ ข	37
4.3 ทำเคลื่อนไหว - พยัญชนะ ค	37
4.4 ทำเคลื่อนไหว - พยัญชนะ ง	37
4.5 แผนภาพชุดข้อมูลของทำแบบมือ (คุณมือแบบที่ 1)	39
4.6 แผนภาพชุดข้อมูลเฉพาะนี้ซึ่งองทำแบบมือ (คุณมือแบบที่ 2)	40

3
บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันมีการพัฒนาอุปกรณ์ต่าง ๆ โดยมีเป้าหมายเพื่อให้ผู้ใช้สามารถใช้งานอุปกรณ์ทางอิเล็กทรอนิกส์โดยใช้วรากษของตนเองเป็นอินพุตได้ และได้รับความรู้สึกสมจริง หรือ สะท้อนความรู้สึกของคนอื่น หากเทียบกับการใช้งานอินพุตปกติอย่างเช่นคีย์บอร์ด, เม้าส์ หรือจอยสัมผัส หากกล่าวถึงอุปกรณ์ที่พยายามตรวจสอบท่าทางของมือ ในปัจจุบันมีการพัฒนาขึ้นมาหลากหลายประเภทเพื่อวัดถูกประสงค์โดยย่างหนีงอย่างชัดเจน ตัวอย่างเช่น ถุงมือตรวจสอบท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality เพื่อใช้ในการเล่นเกมส์เพื่อความบันเทิง หรือฝึกฝนทักษะปฏิบัติเสมือนจริง และถุงมือตรวจสอบท่าทางมือเพื่อใช้แปลงภาษามือเป็นคำหรือตัวสะกดภาษาอังกฤษ เป็นต้น



รูป 1.1 ถุงมือตรวจสอบท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality



รูป 1.2 ถุงมือตรวจสอบท่าทางมือเพื่อใช้แปลงภาษามือ

หากกล่าวถึงภาษาเมืองที่เป็นภาษาไทย เมื่อจากถุงมือตรวจจับการทำทางเมืองเพื่อแปลงเป็นตัวสะกดภาษาอังกฤษตามด้วยชื่อที่กล่าวมาข้างต้นอาจมีการทำต่าง ๆ ที่ใช้เป็นอินพุตรวมถึงเจ้าตู้กดอยู่ในขอนเทศที่ได้กำหนดไว้เพียงเท่านั้น ไม่สามารถนำมาใช้กับภาษาไทยที่มีการทำแตกต่างออกไปจากภาษาเมืองนั้น ๆ ได้ โดยเฉพาะตัวสะกดที่มีมากกว่าอย่างเห็นได้ชัด หรืออาจมีทำที่อยู่นอกเหนือขอนเทศที่ถุงมีจะสามารถตรวจสอบได้ในระดับฮาร์ดแวร์ หรือหากใช้ถุงมือตรวจจับการทำทางสำหรับนักพัฒนาที่อาจต้องมีการพัฒนาต่อโดยใช้ชุดพัฒนาซอฟต์แวร์ที่รองรับจากผู้พัฒนา ร่วมกับความรู้เฉพาะด้าน อิถัทัชจอมราคากลาง ไม่คุ้มค่าที่จะนำมายังผู้พัฒนาที่อ่อนดูประสงค์เดียว ซึ่งเป็นเหตุผลหนึ่งที่ทำให้ควรมีการพัฒนาถุงมือตรวจจับการทำทางสำหรับภาษาเมืองที่เป็นภาษาไทยขึ้นมาเอง หรือถุงมือตรวจจับการทำทางที่สามารถใช้งานในวัสดุประสงค์อื่นได้นอกจากภาษามี

ทางผู้พัฒนาจึงสนใจที่จะสร้างถุงมือเพื่อตรวจจับการทำทางเมือง โดยใช้เทคโนโลยี Machine Learning ที่ได้รับการพัฒนาประสิทธิภาพขึ้นและมีความนิยมในปัจจุบันเข้ามาใช้เพื่อตรวจจับการทำทางเมือง โดยเริ่มจากการทำที่แสดงถึงตัวสะกดไทย เพื่อลดข้อจำกัดในด้านการทำทางที่กล่าวมาข้างต้นลง จะมีแอพพลิเคชันให้ผู้ใช้สามารถตั้งค่าชุดของทำทางที่จะใช้เป็นอินพุตได้ นอกจากนั้น แอพพลิเคชันจะสามารถตั้งค่าการกระทำที่ต้องการจากทำทางที่ตั้งค่าไว้ได้ ตัวอย่างเช่น การแสดงผลรูปแบบแอพพลิเคชัน แสดงข้อความบนจอแสดงผลบนถุงมือ และใช้ประโยชน์จากการเคลื่อนไหว ตัวอย่างเช่น การเคลื่อนไหวเพื่อส่งคำขอ HTTP ไปยังอุปกรณ์อื่นเป็นต้น ดังนั้นถุงมือจะไม่เพียงแค่สามารถตรวจจับการทำที่แสดงถึงตัวสะกดไทยเท่านั้น แต่จะสามารถตรวจจับการทำทางที่เป็นการเคลื่อนไหวได้หลากหลายแบบ หลากหลายทำด้วยตัวของผู้ใช้งาน แต่จะสามารถทำได้มากกว่าที่เคยแค่แสดงผล เพื่อการใช้งานที่หลากหลายวัสดุประสงค์มากขึ้น

1.2 วัสดุประสงค์ของโครงงาน

- 1) พัฒนาถุงมือเพื่อการตรวจจับการทำทางมีรวมถึงระบบที่เกี่ยวข้อง
- 2) ศึกษาการใช้งานไมโครคอนโทรลเลอร์และระบบปฏิบัติการ Linux กายใน
- 3) ศึกษาและพัฒนาแอพพลิเคชันและเซิร์ฟเวอร์เพื่อดึงต่อ กับ Hardware
- 4) ศึกษาการประมวลผลข้อมูลของเซนเซอร์ผ่านกระบวนการทาง Machine Learning

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ถุงมือสามารถจำแนกความสะอาดให้กับผู้ใช้งานได้
- 2) ถุงมือสามารถให้ความบันเทิงกับผู้ใช้งานได้
- 3) ถุงมือสามารถช่วยเป็นสื่อกลางในการสื่อสาร โดยภาษาเมืองได้บางส่วน
- 4) ผู้ใช้สามารถเข้าถึงและตั้งค่าการใช้งานถุงมือได้ง่าย

1.4 ขอบเขตของโครงการ

ถูงมือสามารถตรวจสอบข้อท่าทางในรูปแบบที่เซนเซอร์ 6 ตัวสามารถตรวจสอบได้เท่านั้น ประกอบด้วยเซนเซอร์กล้องหลังมือ 1 ตัว และบนนิ้วแต่ละนิ้ว 5 ตัว การตรวจสอบท่าทางของถุงมือที่เป็นท่านี่ง จะสามารถทำได้อีกอย่างน้อย 20 แบบ สำหรับท่าที่เป็นเคลื่อนไหว สามารถทำได้อีกอย่างน้อย 20 ท่า และโดยรวมแล้ว จะสามารถตรวจสอบท่าทางภาษาไทยที่เป็นตัวสะกดภาษาไทยได้อีกอย่างน้อย 30 ตัว

สำหรับการกระทำหลังจากตรวจสอบท่าทาง เมื่อต้นจะสามารถแสดงผลข้อความบนเว็บแอพพลิเคชันหรือจอแสดงผลได้ และส่งคำขอ HTTP ไปยังอุปกรณ์อื่นในเครือข่ายเดียวกันกับถุงมือได้

1.5 ข้อจำกัดของโครงการ

รูปแบบท่าทางของของมือผู้ใช้ที่ไม่เหมือนกันอาจจะส่งผลให้ความแม่นยำแตกต่างกัน เกิดจากค่าที่ได้จากเซนเซอร์ ซึ่งอาจจะมีค่าผิดพลาดหรือแตกต่างมากพอที่จะทำให้ตรวจสอบได้ท่าที่ผิดพลาดได้ โดยเฉพาะเซนเซอร์ที่ใช้วัดความขององนิ้วมือ ซึ่งวัดการงอได้เพียงแค่ทิศทางที่กำนมือเท่านั้น สำหรับผู้ใช้ที่สามารถยกและหันนิ้วมือไปในทิศทางข้ามได้มาก หรือท่าที่มีความไม่คล่องแยบคาย อาจจะทำให้เกิดการตรวจสอบที่ผิดพลาดได้ แต่ท่าที่อยู่ในกรอบลักษณะนี้ ถือว่าเป็นส่วนน้อยของท่าทั้งหมดที่ถุงมือควรตรวจสอบได้ และไม่ได้อยู่ในขอบเขตการพัฒนา

3

1.6 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

6

หัวข้อกิจกรรม	เดือน									
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ค้นหาหัวข้อที่สนใจ และปรึกษาหัวข้อ ³ ดังกล่าวกับอาจารย์										
2. ศึกษาเทคโนโลยีที่ใช้พัฒนา										
2.1 ศึกษาวิธีการใช้งาน ระบบปฏิบัติการ										
2.2 ทดสอบการติดตั้ง ⁹ ส่วนประกอบต่าง ๆ ที่ใช้ ในการพัฒนา										
3. ออกแบบ										
3.1 ออกแบบโครงสร้าง ฐานข้อมูล ⁹										
3.2 ออกแบบโครงสร้าง ระบบ ⁹										
3.3 ออกแบบโครงสร้าง และวิธีประมวลผล <i>Dataset</i>										
3.4 ออกแบบเว็บ แอพพลิเคชัน										

หัวข้อกิจกรรม	เดือน								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
4. พัฒนา									
4.1 ประกอบถุงมือเพื่อให้สามารถใช้งานได้เบื้องต้น				■					
4.2 พัฒนาเว็บแอพพลิเคชันเพื่อใช้แสดงผลและสร้าง Dataset				■	■				
4.5 พัฒนาเว็บแอพพลิเคชันเพื่อใช้ตั้งค่าอินพุต/เอาต์พุต						■			
5. ทดสอบ ปรับปรุง และแก้ไข									
5.1 ทดสอบและแก้ไขการทำงานโดยรวมของระบบ								■	
5.2 พัฒนาส่วนจ่ายไฟให้ถูกมือ							■		
5.3 ปรับปรุงรูปลักษณ์ถุงมือ								■	
5.4 ปรับปรุงเว็บแอพพลิเคชัน								■	
5.5 ปรับปรุงโมเดล Machine Learning								■	

3
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

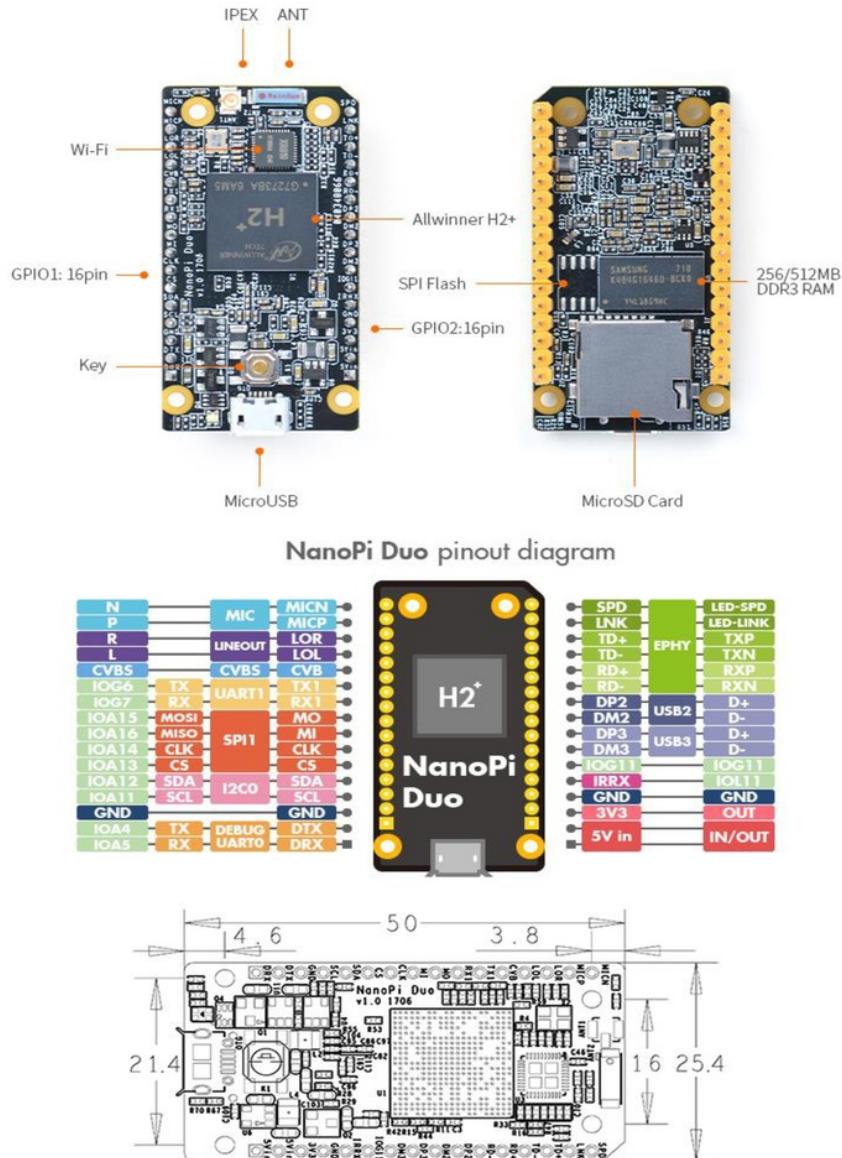
2.1 เทคโนโลยีด้านสาร์ดแวร์

2.1.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

Nanopi-DUO หน่วยประมวลผลขนาดเล็กที่สามารถงานໄว้บนชิปมีอยู่อย่างสะดวก มีคอร์ในการประมวลผลถึง 4 คอร์ จึงสามารถทำงานหลายงานพร้อมกันได้สะดวก เหมาะสมกับโครงการที่จำเป็นจะต้องประมวลผล เปิดเชิร์ฟเวอร์ และติดต่อกับบอร์ดแวร์ในเวลาเดียวกัน อีกทั้งยังรันในระบบปฏิบัติการ Linux (Ubuntu) ที่ยอดนิยม สามารถศึกษาเพิ่มเติมได้ง่าย และสามารถติดตั้งแพคเกจที่จำเป็นต้องใช้ในการดำเนินโครงการได้

ตาราง 2.1 ข้อมูลจำเพาะ Nanopi-DUO [4]

CPU	Allwinner H2+, Quad-core Cortex-A7
RAM	DDR3: 256MB/512MB
การเชื่อมต่อเครือข่าย	10/100M Ethernet, WiFi XR819
อินเทอร์เฟส USB	ขา 2.54มม. pin x2, รวมอยู่ในขา 2.54มม.
ช่องใส่การ์ด MicroSD	x1
MicroUSB	x2: สำหรับ OTG และสำหรับจ่ายไฟ
อินเทอร์เฟสสำหรับ Serial Debugging	รวมอยู่ในขา 2.54มม.
อินเทอร์เฟสสำหรับเสียง	รวมอยู่ในขา 2.54มม.
GPIO1	16 ขา 2.54 มม. ประกอบไปด้วย UART, SPI, I2C, Audio ฯลฯ
GPIO2	16 ขา 2.54 มม. ประกอบไปด้วย USB, 10/100M Ethernet, IO ฯลฯ
ขนาดของ PCB	25.4 x 50 มม.
Power Supply	DC 5V/2A
อุณหภูมิที่วัดได้	-40 ถึง 80 องศาเซลเซียส
ระบบปฏิบัติการ/ซอฟต์แวร์	U-boot, Linux Kernel 4.11.2 (mainline), Ubuntu 16.04.2 LTS (Xenial)

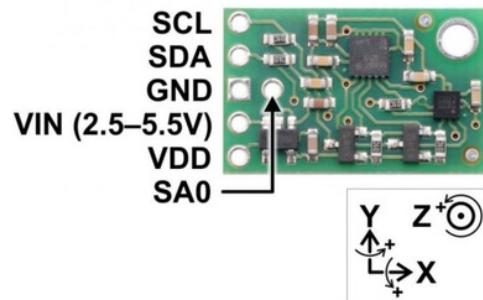


รูป 2.1 แผนภาพโครงสร้าง, ขนาดรวมต่อของ NanoPI-DUO

2.1.2 ເສນເຊອර් (Sensors)

เซนเซอร์ที่ใช้ตรวจจับการเคลื่อนไหวของมือจะประกอบไปด้วยเซนเซอร์วัดค่าความงอ (Flex Sensor) เพื่อวัดความงอของต่อมน้า้มือ และเซนเซอร์ที่วัดความเร็ว (IMU: Inertial Measurement Unit) ซึ่งประกอบด้วยเซนเซอร์วัดความเร่งเชิงเส้น (Accelerometer)

และเซนเซอร์วัดความเร็วเชิงมุม (Gyroscope) เพื่อใช้ตรวจจับการเคลื่อนไหวและวัดมุมปัจจุบันของมือ



รูป 2.2 Polulu MinIMU9-v5

Polulu MinIMU9-v5 เป็นเซนเซอร์หน่วยวัดอิเล็กทรอนิกส์ที่ใช้ในโครงงานนี้ ซึ่งมีขนาดเล็ก วางบนมือและปลายนิ้วได้ สามารถส่งข้อมูลผ่านบัส I2C ได้อีกทั้งยังมีเซนเซอร์สนามแม่เหล็ก (Magnetometer) ซึ่งเป็นเซนเซอร์ตรวจจับทิศทางรวมเข้ามาด้วย กล้ายิ่ง 9 แกน ซึ่งสามารถนำไปประยุกต์ใช้ได้ในอนาคตหากจำเป็น

ตาราง 2.2 ข้อมูลจำเพาะเซนเซอร์ Polulu MinIMU9-v5 [5]

ขนาด:	$0.8'' \times 0.5'' \times 0.1''$
น้ำหนัก:	0.7 g
อินเทอร์เฟส:	I ² C
ตัวชี้ไฟฟ้าต่ำสุดในการทำงาน (Minimum operating voltage):	2.5 V
ตัวชี้ไฟฟ้าสูงสุดในการทำงาน (Maximum operating voltage):	5.5 V
แกนที่วัด:	pitch (x), roll (y), yaw (z)
ช่วงการวัด:	$\pm 125, \pm 245, \pm 500, \pm 1000, \text{ or } \pm 2000^\circ/\text{s}$ ความเร็วเชิงมุม(gyro) $\pm 2, \pm 4, \pm 8, \text{ or } \pm 16$ g ความเร่งเชิงเส้น(accelerometer) $\pm 4, \pm 8, \pm 12, \text{ or } \pm 16$ gauss สนามแม่เหล็ก(magnetometer)
กระแสไฟที่จ่ายออก:	5 mA

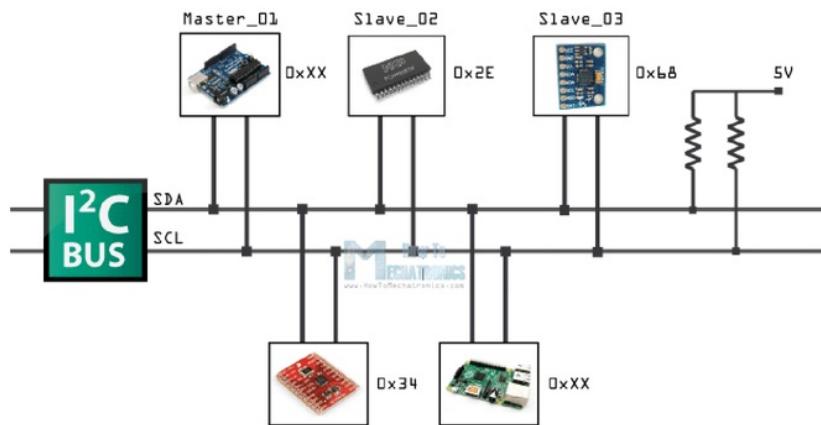
2.1.3 จอแสดงผล (Display)



รูป 2.3 SSD1306

SSD1306 Bi-Color OLED Display เป็นจอแสดงผลสองสี ขนาด 128x64 ชั้น มีขนาดพอดี สามารถวางไว้บนมือได้ นอกจากนั้นยังใช้การสื่อสารด้วย I2C เช่นเดียวกันกับชนิดอื่น ๆ

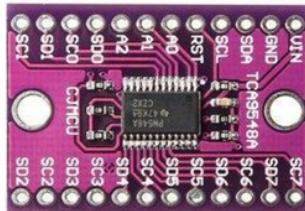
2.1.4 การติดต่อระหว่าง硬件แวร์



รูป 2.4 ตัวอย่างเครือข่ายการเชื่อมต่อของบัส I²C

Inter Integrate Circuit (I2C) เป็นการสื่อสารอนุกรมแบบเข้าสัญญาณ ที่ทึ้งสองฝั่งสามารถเป็นได้ทั้งผู้รับและผู้ส่ง เช่นเซอร์จจะต่อเข้ากับไมโครคอนโทรลเลอร์ผ่านบัส I2C และ

ไมโครคอนโทรลเลอร์สามารถเขียนและอ่านเซนเซอร์ได้จากการเข้าถึงเลขตำแหน่ง (Address) จำเพาะนั้น ๆ ของเซนเซอร์



รูป 2.5 TCA9548A

ไมค์อูลขยายช่องสัญญาณ I²C 8 ช่อง (TCA9548A) เนื่องจากเซนเซอร์หน่วยวัดอาจต้องมีความเรียบง่ายที่ใช้ในการดำเนินโครงการอาจมีหลายตัว แต่ทุกตัวที่ใช้นั้นมีเลขตำแหน่งเดียวกัน เพื่อเก็บไขปัญหาที่เลขตำแหน่งซ้ำ จึงต้องใช้ไมค์อูลตัวนี้ โดยให้หลักการการทำงานเหมือน Multiplexer กล่าวคือสามารถเขียนช่องที่ต้องการไปยังตำแหน่งของ TCA9548A เพื่ออ่านค่าจากช่องนั้น ๆ ได้ สำหรับการอ่านข้อมูลจาก I²C จะใช้ไลบรารี SMBus ของ Python ที่เขียนขึ้นมาเพื่อรับส่งข้อมูลด้วย I²C โดยเฉพาะ

2.2 เทคโนโลยีด้านการประมวลผลท่าทางมือ

การตรวจจับและประมวลผลท่าทางของมือ จะใช้ภาษา Python เวอร์ชัน 3.6 เป็นอย่างมากในการพัฒนา Machine Learning ซึ่งส่งผลให้มีเฟรมเวิร์คหรือไลบรารีต่าง ๆ ที่สนับสนุนมากมาย

TensorFlow เป็นเฟรมเวิร์คที่ใช้ในการพัฒนาโมเดล Machine Learning เพราะเป็นหนึ่งในเฟรมเวิร์คที่นิยมมากในการพัฒนาโมเดล Machine Learning เนื่องจากมีประสิทธิภาพในเรื่องของความเร็วในการคำนวณอย่างเหมาะสมเมื่อเทียบกับความยากในการเขียน และ TensorFlow ถูกพัฒนาโดยบริษัท Google ทำให้มีการอัพเดตเรื่องของประสิทธิภาพของเฟรมเวิร์คนี้อย่างสม่ำเสมอ

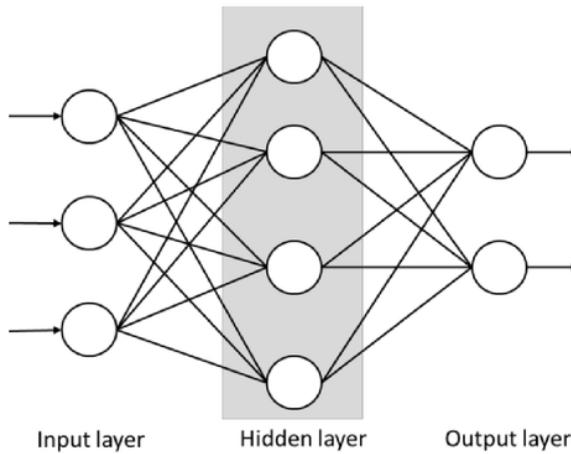


รูป 2.6 สัญลักษณ์ TensorFlow

ไม่เคล็ดสำหรับการพัฒนา Machine Learning เพื่อการทำนายการเคลื่อนไหวของมือ มีทั้งหมด 2 ไม่เคล็ด ได้แก่

2.2.1 โครงข่ายประสาทเทียม (Artificial Neural Networks - ANN)

เป็นไม่เคล็ดทางคณิตศาสตร์ที่จำลองการทำงานของเครื่องข่ายประสาทในสมองมนุษย์ โดยมีวัตถุประสงค์ที่จะสร้างเครื่องมือชี้สิ่งใดความสามารถในการเรียนรู้การจำรูปแบบ (Pattern Recognition) และการสร้างความรู้ใหม่ (Knowledge Extraction) เพื่อเดิมพันความสามารถที่มีในสมองมนุษย์



รูป 2.7 สถาปัตยกรรมโครงข่ายประสาทเทียม

จากรูปที่ 2.7 คือโครงสร้างหลักๆ ของโครงข่ายประสาทเทียม จะมีสามส่วน ได้แก่ ชั้นขาเข้า (Input layer), ชั้นซ่อน (Hidden layer) และชั้นขาออก (Output layer)

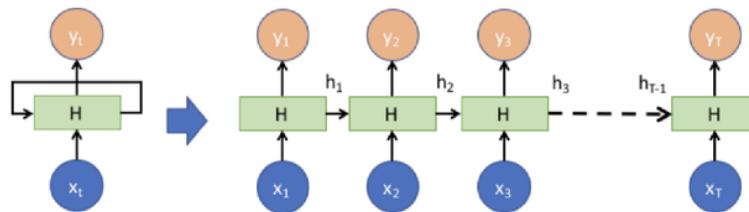
ชั้นขาเข้าจะทำการรับข้อมูลเข้ามาซึ่งอาจมีหลากหลายรูปแบบได้ และชั้นซ่อนจะทำการเอารูปแบบนั้นมาคุณกับค่าน้ำหนักของแต่ละรายละเอียดผลลัพธ์ที่ได้จากทุกๆ ข้อมูลรวมกันแล้วนำมาที่叫做 threshold ที่กำหนดไว้ ถ้าผลลัพธ์เกินกว่า threshold ก็จะทำการส่งผลลัพธ์ออกไป แต่ถ้าต่ำกว่า threshold ก็จะไม่เกิดผลลัพธ์ ซึ่งค่าน้ำหนักจะเป็นค่าที่ไม่แน่นอน แต่สามารถให้คุณพิเศษปรับค่าได้เองโดยการสอนให้คุณพิเศษรู้จัก pattern ของสิ่งที่ต้องการให้คุณพิเศษรู้ นี้叫做 "back propagation" ซึ่งจะมีการใช้อัลกอริทึม back-propagation เพื่อ ใช้ในการปรับปรุงค่าน้ำหนัก หลังจากได้รูปแบบข้อมูลสำหรับฝึกให้ในแต่ละครั้งแล้ว ค่าอัตราศูนย์ที่ได้รับจากเครื่องข่ายจะถูกนำไปเรียนรู้เพื่อบันทึกผลลัพธ์ที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครื่องข่ายเพื่อใช้แก้ไขค่าน้ำหนักต่อไป เมื่อปรับค่าน้ำหนักแล้วจะทำให้มีความแม่นยำมากขึ้น

25

2.2.2 โครงข่ายประสาทเทียมแบบมีหน่วยความจำระยะสั้นและยาว (Long Short-Term Memory)

Long Short-Term Memory (LSTM) เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่ถูกออกแบบมาสำหรับการประมวลผลแบบลำดับ (sequence) และเป็นรูปแบบหนึ่งของ Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) เป็นโครงข่ายประสาทเทียมแบบหนึ่งที่หมายถึงการใช้งานกับข้อมูลที่มีลักษณะเป็นลำดับ (sequence) ซึ่งก็คือการอาคลัสพธ์ที่ได้จากการคำนวณข้อมูลกันมาให้เป็นข้อมูลขาเข้าอีกรัง ดังรูปที่ 13 ซึ่งมีประโยชน์อย่างมากในข้อมูลที่มีความต่อเนื่อง เช่น time series ข้อมูลเสียง ข้อความ เป็นต้น ดังนั้น 2 ส่วนที่สำคัญของ RNN ก็คือ Hidden state ก่อนหน้า และข้อมูลอินพุต ณ ตอนนั้น



Long Short-Term Memory ได้มีการเปลี่ยนตัวฟังก์ชันด้านในให้มีความเสถียรและมีประสิทธิภาพมากขึ้น โดยมีหน่วยความจำเข้ามาเก็บข้อมูลที่ไว้ด้วยจัดการสำเร็จที่เกิดขึ้นและใช้ในการตัดสินใจในเวลาลัดไป โดยในหน่วยความจำนี้จะต้องถูกปรับไปเรื่อยๆ การที่มีหน่วยความจำเข้ามาเก็บข้อมูลนั้นทำให้แก้ปัญหาของ RNN ได้ดีอีกขั้น คือสามารถถูกข้อมูลขอนกันได้ระยะที่ยาวมากขึ้น

2.2.2.1 การทำงานของโครงสร้างประสาทเทียมแบบ LSTM

- 1) Cell state เป็นตัวเก็บ state ของ memory cell ใน LSTM
- 2) Gate เป็นตัวที่ควบคุมการไหลของข้อมูล ซึ่งก็คือ ค่า analog ที่ควบคุมว่าเมื่อไรควรจะ read, write หรือ forget ซึ่งก็หมายความกับประดุจที่จะฉุว่าเมื่อไรควรปิดให้ข้อมูลไหลเข้า ไหลออก หรือไหลหายไป (forget)
 - 2.1) Forget gate คือการลบ cell state ออกไปและเตรียมพื้นที่รับข้อมูลใหม่ โดยมี Forget gate เป็นตัวตัดสินใจ ถ้า Forget gate ให้ค่าเป็น 0 ก็ลบ cell state ออกไป แต่ถ้า Forget gate ให้ค่าเป็น 1 ก็จะบังเก็บ cell state เดิมต่อไป ซึ่งข้อมูลที่จำเป็นต้องใช้ในการตัดสินใจนั้นมาจากข้อมูลอินพุตที่เข้ามาใหม่ ประกอบกับ hidden state ก่อนหน้า โดยจะใช้ sigmoid function เป็นตัวตัดสิน
 - 2.2) Input gate คือตัวที่ตัดสินว่าจะอนุญาตให้อัพเดท cell state หรือไม่ เมื่อมีข้อมูลอินพุตใหม่เข้ามา การคำนวนนี้ใช้ค่าข้อมูลอินพุตที่เข้ามา กับ hidden state ก่อนหน้ากัน
 - 2.3) Input modulation gate คือตัวที่ตัดสินใจว่าต้องการอัพเดท cell state จะอัพเดทด้วยค่าใด
 - 2.4) การอัพเดท cell state คือการนำข้อมูลจาก Forget Gate , Input Gate และ Input modulation Gate มารวมเข้าด้วยกัน ดังนี้

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

กำหนดให้

c_t คือค่าของ cell state

f_t คือค่าของ forget gate

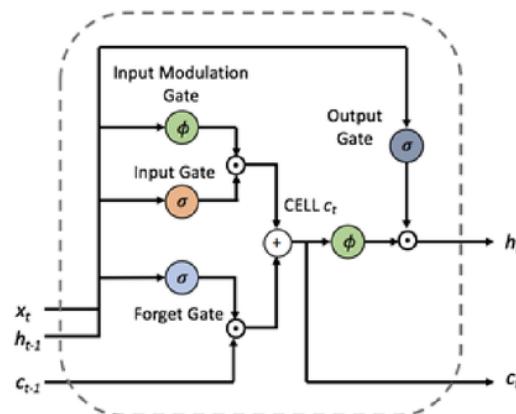
C_{t-1} คือค่าของ cell state ณ เวลาก่อนหน้า

i_t คือค่า input gate

g_t คือค่าที่จะอัปเดท

เริ่มจากส่วนแรกของสมการ ถ้า f_t เป็น 0 ก็จะไม่เอา C_{t-1} มาพิจารณาในการอัปเดท cell state ถ้า f_t เป็น 1 จะยังคงค่า C_{t-1} เอาไว้ประกอบการพิจารณาการอัปเดท ส่วนหลังของสมการ ส่วนนี้จะเป็นส่วนของการอัปเดท cell state จากข้อมูลใหม่ ซึ่งตอนนี้มีค่าที่จะอัปเดตจาก g_t แล้ว และจะใช้ g_t มาเป็นตัวตัดสินในการอัปเดท ถ้า g_t เป็น 1 ก็อัปเดตได้เลย แต่ถ้าเป็น 0 ก็จะไม่มีการอัปเดต จากค่าทั้งหมดทำให้ได้ค่า c_t ตัวใหม่

2.5) Output gate คือสิ่งที่ต้องผลิตออกไป ก็คือ hidden state ณ เวลาที่ t หรือ h_t ซึ่งเมื่อตอนที่เวลา $t+1$ ตัว LSTM จะเอาค่า h_t นี้ไปคำนวณด้วย ซึ่งการส่งค่า h_t ออกไประบันจะมีตัว Output Gate เป็นตัวตัดสินใจ และจะนำค่าเอาค่าพุดไปเป็นค่า h_t (อินพุต) สำหรับ sequence ต่อไป จะเห็นได้ว่า LSTM จะหมายกับข้อมูลที่เข้ามาเป็นลำดับและสามารถนำข้อมูลเก่ามาคำนวณซึ่งอาจจะมีผลต่อในอนาคตได้ ไม่คล็นน์จึงหมายที่จะนำมาทำงานคุณภาพอากาศในอนาคตเนื่องจากข้อมูลที่เข้ามาเป็นข้อมูลที่เป็นลำดับและมีความต่อเนื่อง [6]



รูป 2.9 สถาปัตยกรรมโครงร่างข่ายประสาทเทียมแบบ LSTM

2.3 เทคโนโลยีด้านฐานข้อมูล

2.3.1 MongoDB



รูป 2.10 สัญลักษณ์ MongoDB

เป็นระบบจัดการฐานข้อมูลแบบไม่เชิงสัมพันธ์ (Non-relational Database) ที่ยอดนิยม จุดเด่นคือสามารถเก็บข้อมูลเป็นประเภทต่ำๆ ที่ไม่จำเป็นต้องมีขนาดคงที่ได้ซึ่งมีประโยชน์อย่างมากในการดำเนินโครงการงาน เนื่องจากมีข้อมูลบางชนิดที่ไม่ทราบขนาด การใช้ฐานข้อมูลแบบนี้และเก็บข้อมูลเป็นวัสดุแทนจะทำให้ดำเนินการเก็บและเรียกคืนได้ง่ายขึ้น

2.4 เทคโนโลยีด้านแอพพลิเคชัน

ในการพัฒนาแอพพลิเคชันโดยตอนกับผู้ใช้ ผู้พัฒนาเลือกที่จะทำเป็นเว็บแอพพลิเคชัน เนื่องจากเมื่อผู้ใช้ชื่อมต่อ กับถุงมือ ผู้ใช้ก็จะสามารถเข้าถึงได้ทันทีผ่านอุปกรณ์ทั่วไป เช่น คอมพิวเตอร์, โทรศัพท์ และแท็บเล็ต โดยไม่ต้องดาวน์โหลดแอพพลิเคชันใด ๆ ล่วงหน้า

2.4.1 Vue.JS



รูป 2.11 สัญลักษณ์ Vue.JS

เป็นเฟรมเวิร์กที่ใช้ในการพัฒนาเว็บแอพพลิเคชันผู้ใช้ เขียนโดยใช้ภาษา Javascript โดยเป็นเฟรมเวิร์กสามารถพัฒนาเว็บแอพพลิเคชันที่สามารถเปลี่ยนแปลงข้อมูลได้แบบเรียลไทม์ จึงเหมาะสมสำหรับการพัฒนาที่ต้องมีการป้อนและอัพเดทข้อมูลจากเซนเซอร์ตลอดเวลา

2.4.2 Flask

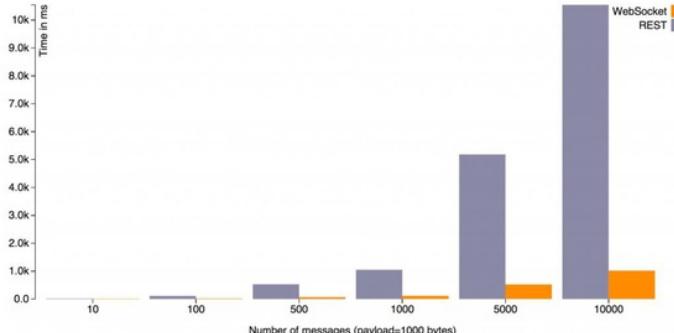


รูป 2.12 สัญลักษณ์ Flask

Flask เป็นเฟรมเวิร์กขนาดเล็กที่ใช้ในการพัฒนาเว็บแอปพลิเคชันผ่านเซิร์ฟเวอร์ เกี่ยนโดยใช้ภาษา Python สามารถติดต่อฐานข้อมูล MongoDB ได้ง่าย และด้วยเนื่องจากการประมวลผลข้อมูลทำได้โดยใช้ภาษา Python จึงควรใช้เฟรมเวิร์กสำหรับพัฒนาเซิร์ฟเวอร์เป็นภาษาเดียวกันด้วย ถึงจะสามารถทำงานร่วมกันได้

2.4.3 Socket.IO

Socket.IO เป็นไลบรารีที่ทำให้เว็บแอปพลิเคชันผู้ใช้สามารถเชื่อมต่อกับเซิร์ฟเวอร์แบบเรียลไทม์ได้ ผ่านเทคโนโลยี WebSocket เมื่อมีการเชื่อมต่อตั้งแต่เริ่มเข้าเว็บ แอปพลิเคชัน ทั้งสองฝ่ายสามารถส่งข้อมูลหากันได้ตลอดเวลา และใช้เวลาอ่อนบ้าง เพราะไม่จำเป็นจะต้องเปิดการเชื่อมต่อทุกครั้งที่ส่งข้อมูลดังเช่นการส่งข้อมูลผ่าน HTTP อย่าง REST ที่นิยมทั่วไป [7]



รูป 2.13 แผนภูมิเปรียบเทียบเวลาที่ใช้ในการส่งข้อมูลระหว่าง REST กับ WebSocket

2.4.4 Eventlet

เป็นไลบรารีสำหรับช่วยในการทำงานแบบ Multithread ช่วยให้สามารถทำการเปิดเซิร์ฟเวอร์โดยใช้ Flask และรอข้อมูลจาก Socket.IO พร้อมกับการประมวลผลข้อมูลด้วย Machine Learning ไปพร้อม ๆ กันได้

2.5 ทฤษฎีเกี่ยวกับการคำนวณ

2.5.1 การคำนวณท่านมุมโดยใช้เซนเซอร์วัดความเร่งเชิงเส้น

เซนเซอร์วัดความเร่งเชิงเส้น สามารถคำนวณมุมได้สองแบบ ดังนี้ [8]

$$A_x = \tan^{-1} \left(\frac{X}{\sqrt{Y^2 + Z^2}} \right)^{23}$$

$$A_y = \tan^{-1} \left(\frac{Y}{\sqrt{X^2 + Z^2}} \right)$$

กำหนดให้

Ax คือมุมแกน x ที่คำนวณได้

Ay คือมุมจากแกน y ที่คำนวณได้

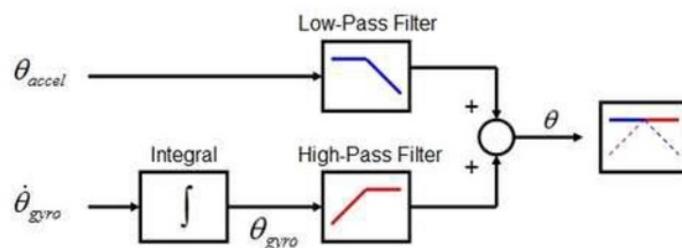
X คือค่าความเร่งแกน x

Y คือค่าความเร่งแกน y

Z คือค่าความเร่งแกน z

แต่เนื่องจากเซนเซอร์วัดความเร่งเชิงเส้นไม่สามารถแยกแยะความเร่งพื้นฐานของ
วัตถุกับความเร่งที่ภายนอกกระทำกับวัตถุได้ จึงอาจจะทำให้เกิดมุมที่ผิดพลาด ให้หากมีแรงภายนอก
มากระทำ หรือมีการขับที่เร็วเกินชั้น

2.5.2 การคำนวณท่านมุมโดยใช้ตัวกรองร่วมสองแบบ (Complementary filter)



รูป 2.14 ระบบของตัวกรองร่วมสองแบบ

เป็นขั้นตอนการคำนวณหามุมของวัตถุ โดยการใช้ค่าที่ได้จากเซนเซอร์วัดความเร่ง เชิงเส้นกับเซนเซอร์วัดความเร็วเชิงมุมร่วมกันเพื่อสร้างเป็นตัวกรอง

ค่าที่ได้จากเซนเซอร์วัดความเร่ง เชิงเส้นจะถูกนำไปคำนวณโดยเป็นมุมโดยใช้สูตร การคำนวณตามข้อที่ 2.5.1 เรียกว่าตัวกรองช่วงล่าง (Low-pass filter)

ค่าที่ได้จากเซนเซอร์วัดความเร็วเชิงมุมจะถูกนำไปคำนวณมุมที่ภาชนะไปได้แนบตั้งแต่ การคำนวณครึ่งก่อนหน้า เรียกว่าตัวกรองช่วงบน (High-pass filter)

จากนั้นจึงนำมาคำนวณตามแต่ละแกน โดยใช้สูตรดังกล่าว [9]

$$\text{Angle}_{axis} = \alpha * (\text{Angle}_{acc}) + (1 - \alpha) * (\text{Angle}_{gyr})$$

กำหนดให้

Angle_{axis} คือมุมผลลัพธ์ ที่ได้จากการคำนวณของมุมนั้น ๆ

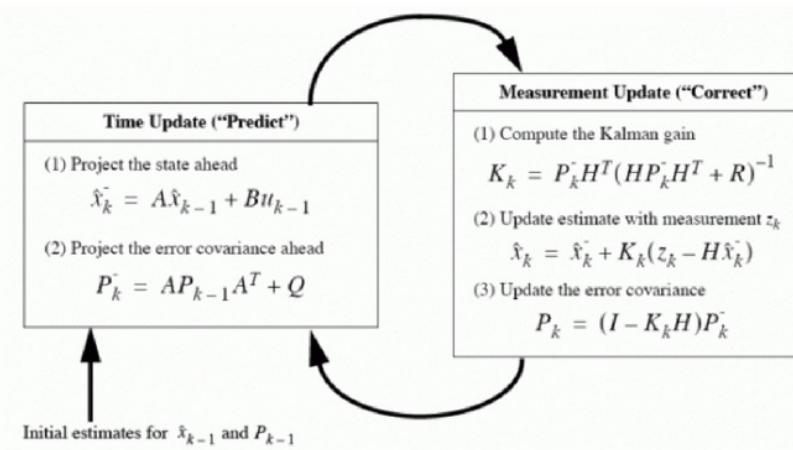
α คือค่าน้ำหนักของ Low-Pass Filter มีค่าระหว่าง 0.0 ถึง 1.0

Angle_{acc} คือค่ามุมที่ได้จากเซนเซอร์วัดความเร่งเชิงเส้น

Angle_{gyr} คือค่ามุมที่ภาชนะได้จากเซนเซอร์วัดความเร็วเชิงมุม

ข้อดีที่เห็นได้ชัดของขั้นตอนการคำนวณแบบนี้ คือสามารถคำนวณได้เร็ว ใช้ทรัพยากรน้อย ให้ค่าที่แม่นยำ แต่ข้อเสียคือไม่สามารถกำจัดค่ารบกวนได้ทั้งหมด

2.5.3 การคำนวณหามุมโดยใช้ตัวกรองคอลมาnan (Kalman filter)



รูป 2.15 ระบบของตัวกรองค่าความ

ความหมายของตัวแปร โดยสังเขป

- x = ค่าสถานะที่คาดเดา (Estimate State)
- A = เมตริกซ์ที่เกี่ยวกับการเปลี่ยนแปลงสถานะ (State transition matrix)
- B = เมตริกซ์ที่เกี่ยวกับตัวแปรควบคุม (Control matrix)
- u = ค่าของตัวแปรควบคุม (Control Signal)
- P = ความแปรปรวนของข้อมูลที่คาดเดา (Error covariance)
- Q = ค่าความผิดพลาดในขั้นตอนการคำนวณ (Estimated error in process)
- K = Kalman gain
- H = เมตริกซ์ที่เกี่ยวกับจำนวนตัวแปรในระบบ (Observation matrix)
- R = ค่าความผิดพลาดของค่าสถานะจริง (Estimated error in measurements)
- z = ค่าสถานะจริงจากการวัด (อินพุตจากเซนเซอร์)

Kalman filter จะมี 5 ขั้นตอนคือ

- 1) การคาดเดาสถานะถัดไปค่าจากตัวแปรควบคุม โดยการคำนวณความเร็วและ
ระยะทางจากเซนเซอร์ รับความเร่งเชิงเส้นและเซนเซอร์ความเร็วชิจมุนโดยใช้
สูตรคำนวนหาความเร็วจากความเร่งและระยะทางจากความเร็ว
- 2) การคำนวณค่าความผิดพลาดของความแปรปรวนของข้อมูล (Error covariance)
ที่จะใช้ในขั้นตอนต่อไป ซึ่งจะส่งผลต่อค่า K (Kalman Gain)
- 3) การคำนวณค่า Kalman Gain จากค่า Error covariance

- 4) การปรับค่าหลังจากได้ค่าที่แท้จริง หลังจากการการคาดเดาในขั้นตอนที่ 1 แล้ว ในขั้นตอนนี้คือการรับค่าอินพุตจากด้านนอกเข้ามาคำนวณด้วยเพื่อความแม่นยำมากขึ้น โดยอินพุตดังกล่าวมีค่าสถานะที่วัดได้จริง
- 5) การปรับค่า Error covariance เพื่อใช้ในการคำนวณในรอบต่อไป (การประมวลผลอย่างต่อเนื่อง) [10]

2.6 งานวิจัยที่เกี่ยวข้อง

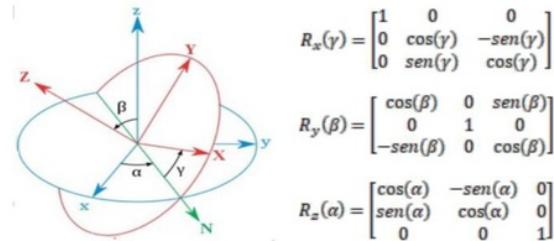
2.6.1 A Gesture Detection Glove For Human-computer Interaction [1]

งานวิจัยนี้มีป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับท่าทางต่าง ๆ ในการควบคุมมาส์และคีย์บอร์ดผ่านการเชื่อมต่อแบบ Bluetooth โดยใช้ MCU เป็น PAM71UM ที่ประกอบจาก FPGA พัฒนาโดยมหาวิทยาลัย Pampa และใช้เซนเซอร์หน่วยวัดอักษรหลักความเร็ว ที่กลางมือและปลายนิ้วทั้ง 5 รวม 6 ตัว



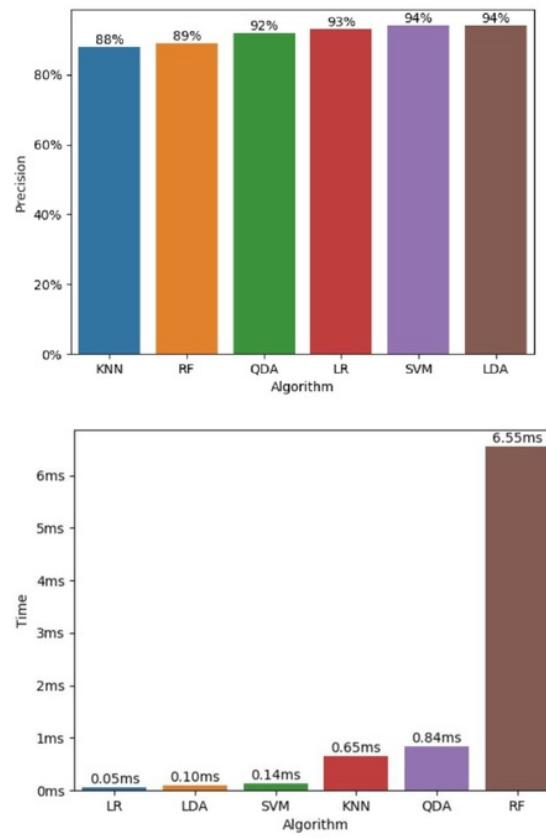
รูป 2.16 ถุงมือของงานวิจัย A Gesture Detection Glove For Human-computer Interaction

ค่าที่ได้จากเซนเซอร์หน่วยวัดอักษรหลักความเร็วอย่างเส้นจะถูกปรับให้อยู่ในค่าระหว่าง -10 ถึง +10 และเซนเซอร์หน่วยวัดอักษรหลักความเร็วอย่างมุมจะถูกปรับให้อยู่ในค่าระหว่าง $-\pi$ ถึง $+\pi$ จากนั้นจะถูกนำมาแปลงเป็นเมตริกซ์การหมุน (Rotation Matrix)



รูป 2.17 Rotation Matrix

การตรวจสอบท่าทางมีจะเป็นการพัฒนาและใช้โมเดล Machine Learning หลากหลายแบบ ได้แก่ KNN, RF, QDA, LR, SVM และ LDA เพื่อนำมาปรับเปลี่ยนความแม่นยำ และเวลาที่ใช้

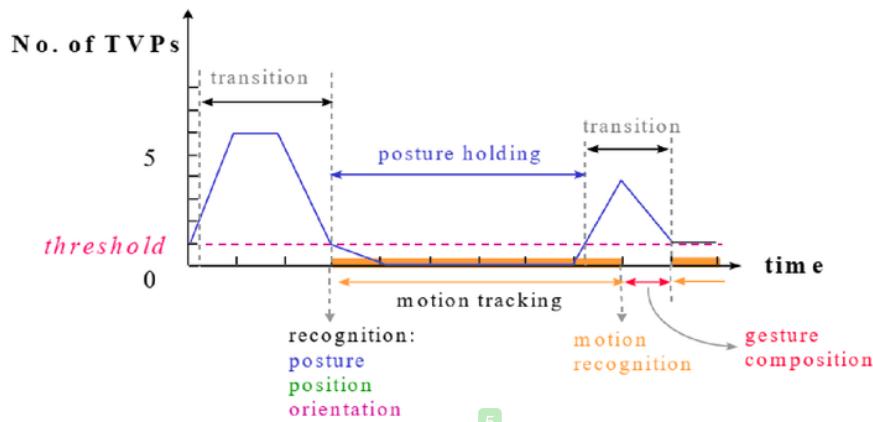


รูป 2.18 แผนภูมิเบริชบ์ที่ยืนยันความแม่นยำและเวลาที่ใช้ระหว่างโมเดล Machine Learning แต่ละตัวของงานวิจัย A Gesture Detection Glove For Human-computer Interaction

5

2.6.2 A real-time continuous gesture recognition system for sign language [2]

งานวิจัยนี้มีเป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับท่าทางของมือ 40 ท่า เพื่อแบ่งเป็นชั้พท์ตามภาษา手语อเมริกัน โดยการหาคุณลักษณะจากชุดข้อมูลชนเชอร์ตามเวลา นำเข้าไปฝึกในโมเดลโครงข่ายประสาทเทียมแบบ Markov Model



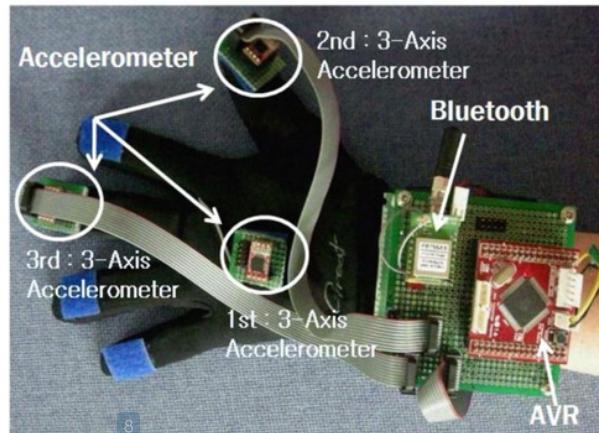
5

รูป 2.19 แผนภูมิเวลาในการวิเคราะห์ท่าของงานวิจัย A real-time continuous gesture recognition system for sign language

8

2.6.3 3-D hand motion tracking and gesture recognition using a data glove [3]

งานวิจัยนี้มีเป้าหมายเพื่อประดิษฐ์ถุงมือเพื่อตรวจจับการเคลื่อนไหวของมือและท่าทางต่าง ๆ โดยใช้ MCU เป็น ATmega128 (AVR) ใช้เซนเซอร์เซนเซอร์หน่วยวัดอัศยหลักความเร็วที่กลางมือ, ปลายนิ้วโป้ง และปลายนิ้วกาง รวม 3 ตัว และติดต่อกันคอมพิวเตอร์ผ่านสัญญาณ Bluetooth



รูป 2.20 ถุงมือจากนวัตกรรม 3-D hand motion tracking and gesture recognition using a data glove

การตรวจจับการเคลื่อนไหว จะใช้การเทียบค่าจากเซนเซอร์กับมุมต่าง ๆ ของมือ ส่วนการตรวจจับท่าทาง จะใช้การตรวจสอบเงื่อนไขว่าหากค่าตรงกับเงื่อนไขของท่าใด ก็จะ ตรวจจับได้ตรงกับท่านั้น ๆ การทดสอบของงานนี้ยืนพบร่วมจากการทดสอบท่าละ 50 ครั้ง สามารถแยกท่าเป็นอิสระ 3 ท่า ได้แก่ ห้อน กระชาย กระไก ออกจากกันได้ร้อยละ 100

บทที่ 3

การออกแบบและพัฒนา

3.1 การออกแบบระบบโดยรวม

28

เพื่อให้เห็นได้ชัดเจน ระบบสามารถแบ่งออกได้เป็น 3 ส่วน คือ ฮาร์ดแวร์, เชิร์ฟเวอร์ และเว็บแอปพลิเคชัน

3.1.1 ฮาร์ดแวร์ (Hardware)

คือส่วนของกลุ่มมือทั้งหมด รวมถึงเซนเซอร์และ MCU มีหน้าที่เก็บตัวอย่างข้อมูล (Sample) จากการทำท่าทางต่าง ๆ ของผู้ใช้ และแสดงผลข้อมูลต่าง ๆ ตามที่ได้คำสั่งจากเชิร์ฟเวอร์

3.1.2 เชิร์ฟเวอร์ (Server)

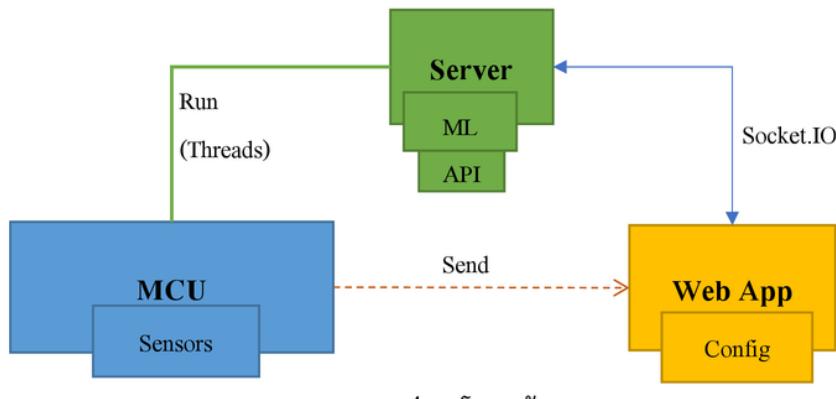
คือส่วนซอฟต์แวร์ที่รับบัน MCU ตลอดเวลา เขียนด้วยภาษา Python และมีหน้าที่ดังต่อไปนี้

- อ่านข้อมูลจากฮาร์ดแวร์ ทำการคำนวณเพื่อแปลงข้อมูลหรือเพื่อความแม่นยำหากจำเป็น
- ตรวจสอบข้อมูลจากฮาร์ดแวร์เพื่อหาช่วงของชุดข้อมูลที่ต้องนำไปคำนวณในโมเดล Machine Learning
- นำชุดข้อมูลไปคำนวณในโมเดล Machine Learning และรอรับคำตอบ
- ส่งเว็บแอปพลิเคชันให้ผู้ใช้หากมีการเรียกใช้งาน
- คอบรับและตอบคำร้องขอข้อมูลผ่าน HTTP (HTTP Request) จากเว็บแอปพลิเคชันผู้ใช้
- คอบรับและส่งข้อมูลผ่าน WebSocket จากเว็บแอปพลิเคชันผู้ใช้

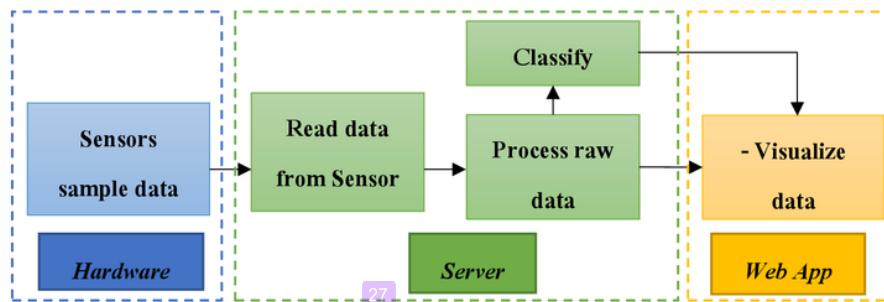
3.1.3 เว็บแอปพลิเคชัน (Web Application)

คือส่วนที่จะทำงานกีต่อเมื่อผู้ใช้ส่งคำขอ HTTP (HTTP Request) ให้แก่เชิร์ฟเวอร์ แล้วเชิร์ฟเวอร์ก็จะส่งคำตอบรับ (HTTP Response) เป็นเว็บแอปพลิเคชัน และทำงานอยู่บนブラัวเซอร์ของผู้ใช้งาน ส่วนนี้มีหน้าที่เบื้องต้นดังนี้

- เชื่อมต่อ รับ และส่งข้อมูลผ่าน WebSocket จากเชิร์ฟเวอร์
- แสดงผลข้อมูล/ผลการตรวจจับท่าทางที่ส่งมาจากเชิร์ฟเวอร์
- เป็นส่วนที่คิดต่อ กับผู้ใช้ (UI) สำหรับการตั้งค่าต่าง ๆ ที่เกี่ยวกับระบบโดยรวม



รูป 3.1 โครงสร้างระบบ



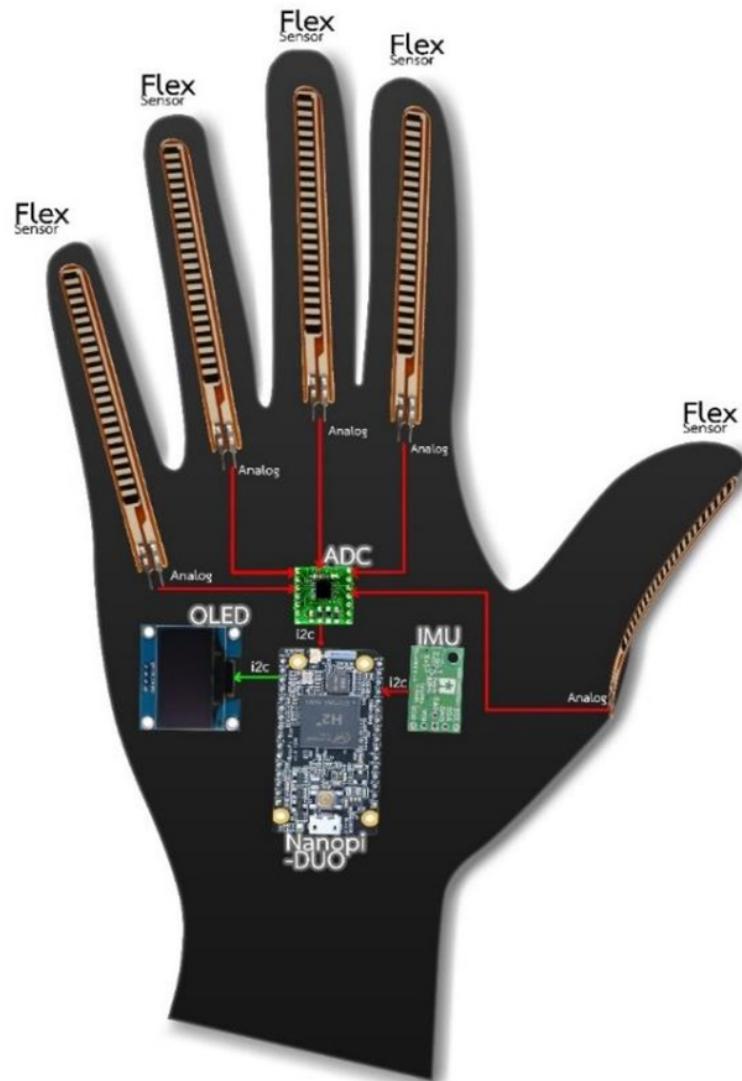
รูป 3.2 ขั้นตอนการทำงานแบ่งตามส่วนของการตรวจสอบท่าทาง

3.2 การออกแบบอุปกรณ์

สำหรับอุปกรณ์ ผู้พัฒนาได้ออกแบบไว้ 2 แบบ คือแบบที่ใช้เซนเซอร์วัดความงอสำหรับนิ้ว และเซนเซอร์หน่วยวัดอัตราศักยหลักความเร็วของสำหรับวัดมุมของมือ และอีกแบบคือแบบที่ใช้เซนเซอร์หน่วยวัดอัตราศักยหลักความเร็วของทั้งความงอและมุมของมือ ถ้ามีทั้ง 2 แบบ จะมีรูปแบบข้อมูลที่ได้รับแตกต่างกัน และส่งผลให้การประมวลผลในโมเดล Machine Learning ต่างกันด้วย

3.2.1 ถุงมือแบบที่ 1:

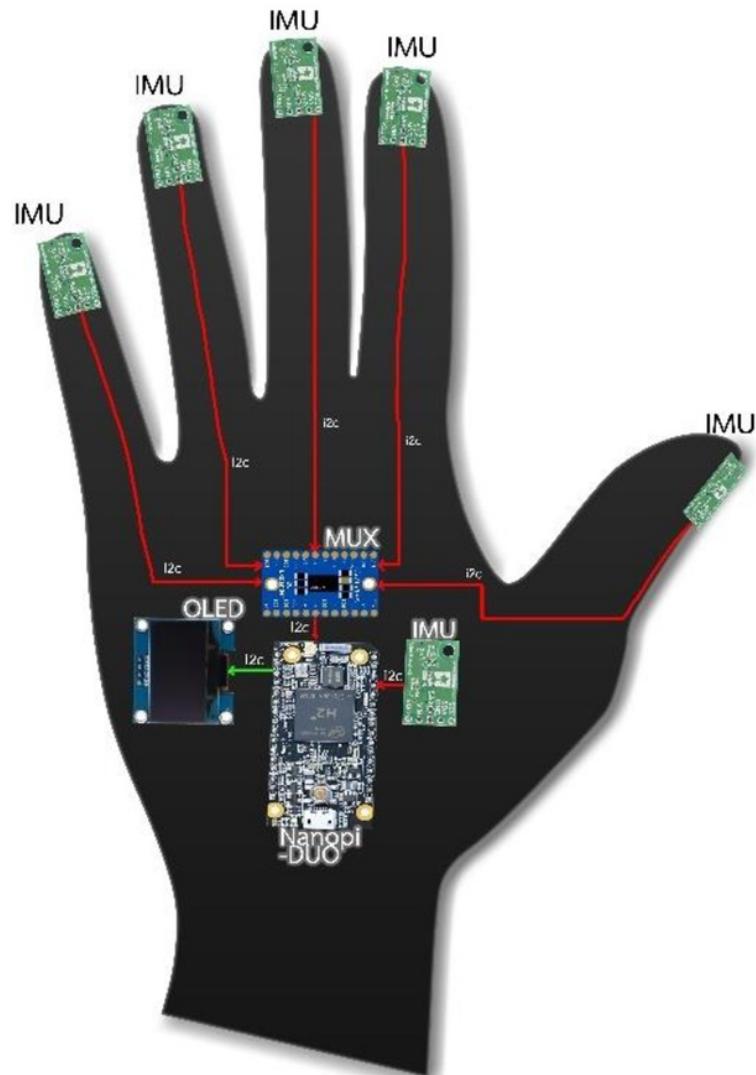
เป้าหมายของถุงมือรุปแบบนี้ คือเพื่อความแม่นยำของการอ่านนิ้ว 1 แกนที่เป็นแกนหลัก นั้นคือแกนในการกำมือ และเพื่อความง่ายในการพัฒนา เช่นเซอร์ที่ใช้จะมีทั้งหมด 6 ตัว ประกอบไปด้วยเซ็นเซอร์วัดค่าความงอของนิ้วนิ้ว 5 ตัว วางแผนนิ้วนิ้วเด่นนิ้ว และเซ็นเซอร์หน่วยวัดอ่างข้อหลักความเรื่อยตรวจสอบการเคลื่อนไหวและการวางแผนมือ วางแผนกลางหลังมือ 1 ตัว



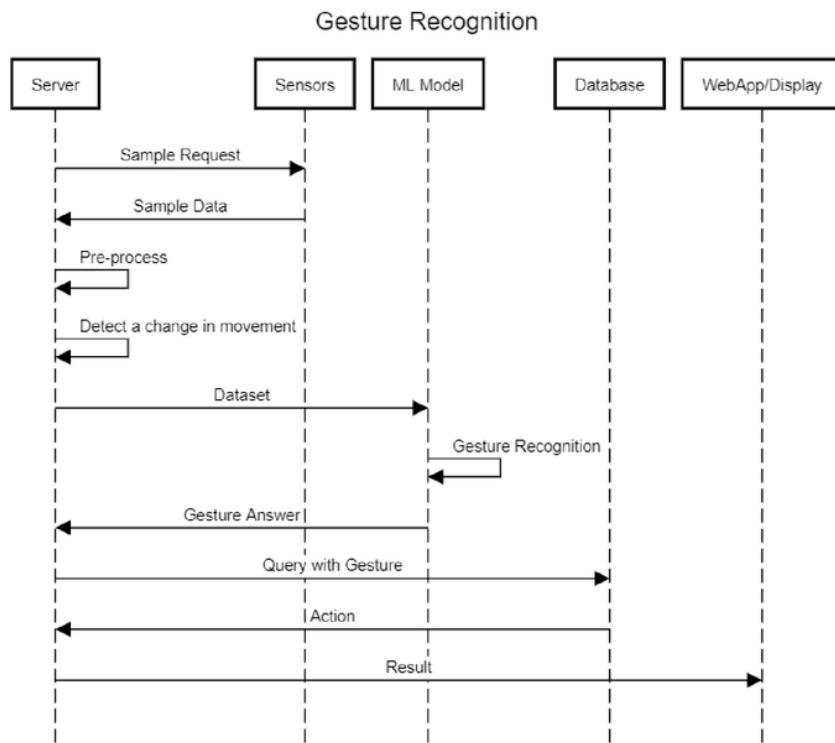
รูป 3.3 โครงสร้างถุงมือแบบที่ 1

3.2.2 ถุงมือแบบที่ 2:

เป้าหมายของถุงมือรุปแบบนี้ คือเพื่อความแม่นยำของตำแหน่งนิ้วในทุกแgn
เซนเซอร์ที่ใช้มีทั้งหมด 6 ตัว ประกอบไปด้วยเซนเซอร์หน่วยวัดอัตราศักดิ์ความเร็วของนิ้ว 5
ตัว วางแผนปุ่มนิ้วนิรภัยเลื่อนของเดลล์นิ้ว และเซนเซอร์หน่วยวัดอัตราศักดิ์ความเร็วของทุกส่วน
การเคลื่อนไหวและการวางแผนมือ วางไว้กลางหลังมือ 1 ตัว [1]



รูป 3.4 โครงสร้างถุงมือแบบที่ 2



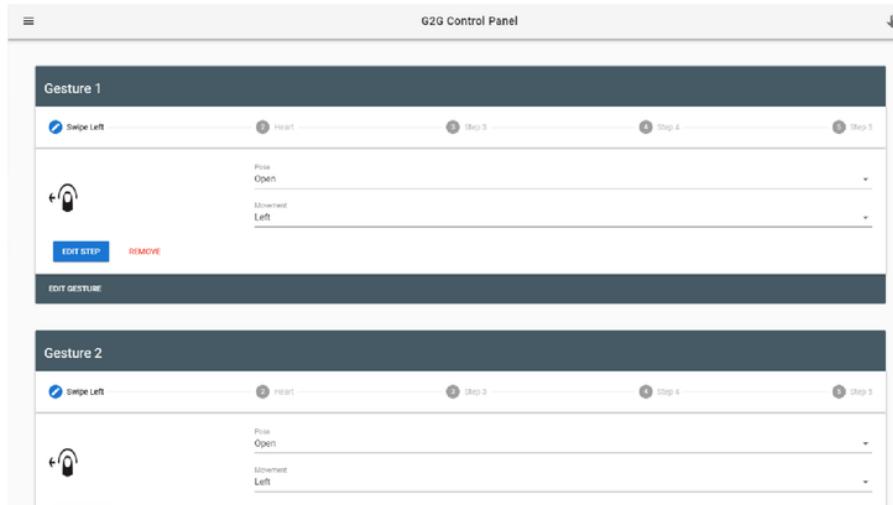
รูป 3.5 แผนผังลำดับการทำงานของการตรวจสอบท่าทาง

3.3 การออกแบบเว็บแอพพลิเคชัน

เว็บแอพพลิเคชันจะแบ่งออกเป็นส่วนหลัก ๆ 2 ส่วน คือส่วนของผู้ใช้ และส่วนของนักพัฒนา

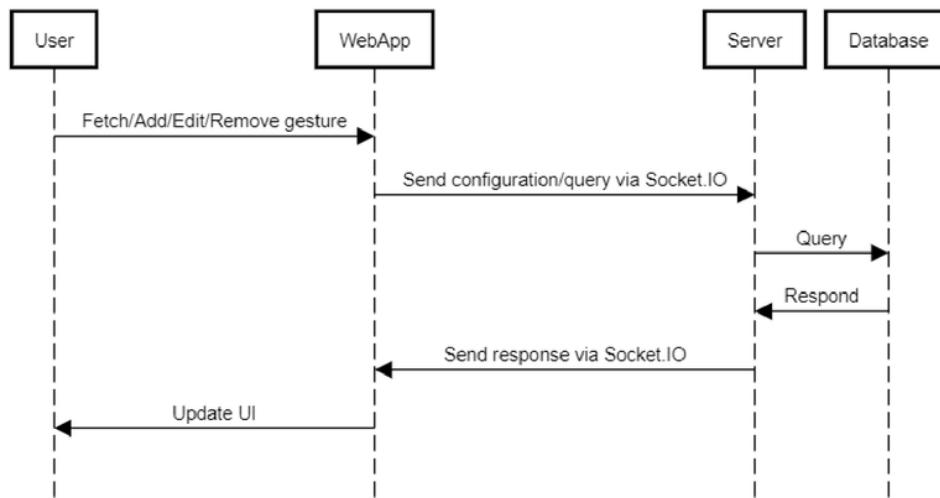
3.3.1 ส่วนของผู้ใช้

มีจุดหมายเพื่อให้ผู้ใช้งานถูมีตั้งค่าท่าทางต่าง ๆ ว่าต้องการทำท่าแบบใดและต้องการให้แสดงผลหรือเกิดสิ่งใดขึ้น โดยจะมีการแสดงรายการท่าทางและขั้นตอนในการทำท่าทางนั้น ๆ เรียกว่า ผู้ใช้สามารถเพิ่ม แก้ไข ลบขั้นตอนต่าง ๆ ได้



รูป 3.6 ตัวแบบหน้าตาเว็บแอพพลิเคชันส่วนของผู้ใช้

Gesture Fetch/Configuration



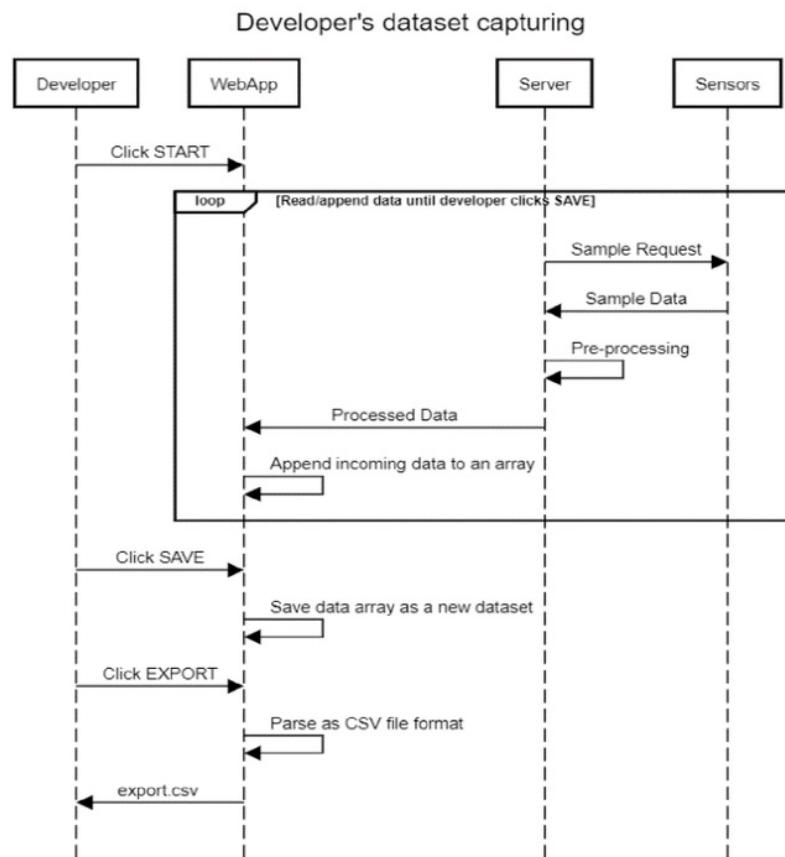
รูป 3.7 แผนผังลำดับการทำงานของการเรียกดู/เพิ่ม/แก้ไข/ลบ ท่าทางของผู้ใช้

3.3.1 ส่วนของนักพัฒนา

มีจุดหมายเพื่อแสดงผลข้อมูลจากเซนเซอร์ และแสดงผลข้อมูลจากฐานข้อมูล ทดสอบส่วนต่างๆ ของระบบ บันทึกชุดข้อมูลสำหรับสร้างโมเดล Machine Learning และรวมถึงสิ่งที่มีประโยชน์ต่างๆ ที่ใช้ในกระบวนการพัฒนา



รูป 3.8 ต้นแบบหน้าตาเว็บแอพพลิเคชันส่วนของนักพัฒนา



รูป 3.9 แผนผังลำดับการทำงานของการบันทึกและส่งออกข้อมูลของนักพัฒนา

3.4 การออกแบบฐานข้อมูล

มีเป้าหมายสำหรับเก็บการตั้งค่าของผู้ใช้ จะใช้ Collection ทั้งหมด 4 Collection ประกอบไปด้วย

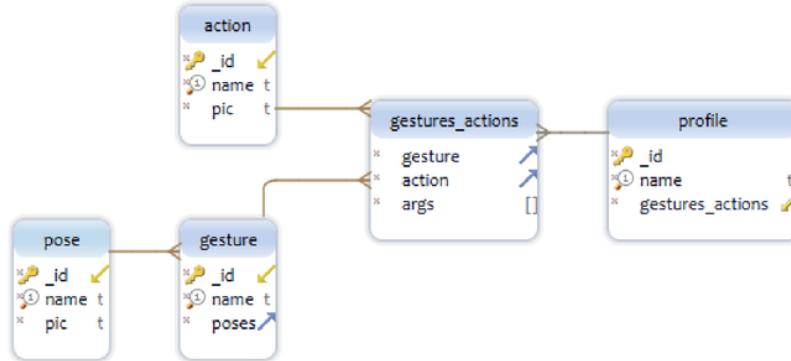
Pose: ท่านั่งหรือท่าขับ 1 ท่า เส้น กำเมือง

Gesture: ชุดท่าทาง 1 ชุด ประกอบไปด้วย pose หลายท่าที่ทำต่อเนื่องกัน เส้น แบบมีเส้นเดียวคือทำมือ

Action: การกระทำหลังจากตรวจจับท่าได้ 1 การกระทำ เส้น แสดงผลข้อความบนจอแสดงผล

Gestures_actions: ประกอบไปด้วย gesture และ action อย่างละเอียด และ args เพื่อกำหนดค่าที่จำเป็นต้องใช้เพิ่มเติมในการกระทำ เส้น ทำมือจะแสดงผลข้อความตามค่าที่เก็บใน args

Profile: โปรไฟล์ 1 โปรไฟล์ คือชุดของท่าและการกระทำ เส้น โปรไฟล์สำหรับตรวจจับและแสดงผลภาษาไทย และโปรไฟล์สำหรับใช้ภาษาอังกฤษ



รูป 3.10 แผนภาพฐานข้อมูล

3.5 การออกแบบชุดข้อมูล

ชุดข้อมูลที่จะส่งไปใช้ในการฝึกโมเดลและการตรวจจับท่าทาง จะเป็นชุดของค่าที่จับได้จากเซนเซอร์ติดต่อกัน 20 ค่า ที่แสดงการเปลี่ยนแปลงของค่าเซนเซอร์ติดต่อกันภายในเวลา 20 วินาที จึงทำให้ชุดข้อมูลหนึ่งจะมี 20 ชุดย่อย แต่โครงสร้างของชุดย่อยจะแตกต่างกันไปตามรูปแบบของกลุ่มมือทั้งสองแบบดังนี้

3.5.1 การออกแบบชุดข้อมูลของอุปกรณ์แบบที่ 1

ชุดย่อยจะประกอบไปด้วยค่ามุมจากเซนเซอร์ 3 แกน และค่าจากเซนเซอร์วัดความจอดตามนี้ 5 นิ้ว รวมเป็น 8 ค่า ชุดย่อยสามารถแสดงเป็นอาร์เรย์ได้ดังนี้

[มือ_x, มือ_y, มือ_z, โป้ง, ซี่, กลาง, นาง, ก้อย]

3.5.2 การออกแบบชุดข้อมูลของอุปกรณ์แบบที่ 2

ชุดข้อมูลประกอบไปด้วยค่ามุนจากเซนเซอร์ เป็นต้นจะใช้ทั้ง 3 แกน รวมเป็น 18 ค่า ชุดข้อมูลสามารถแสดงเป็นอาร์เรย์ได้ดังนี้ [มือ_x, มือ_y, มือ_z, ปี๊ง_x, ปี๊ง_y, ปี๊ง_z, ชี้_x, ชี้_y, ชี้_z, กลาง_x, กลาง_y, กลาง_z, นาง_x, นาง_y, นาง_z, ก้อย_x, ก้อย_y, ก้อย_z]

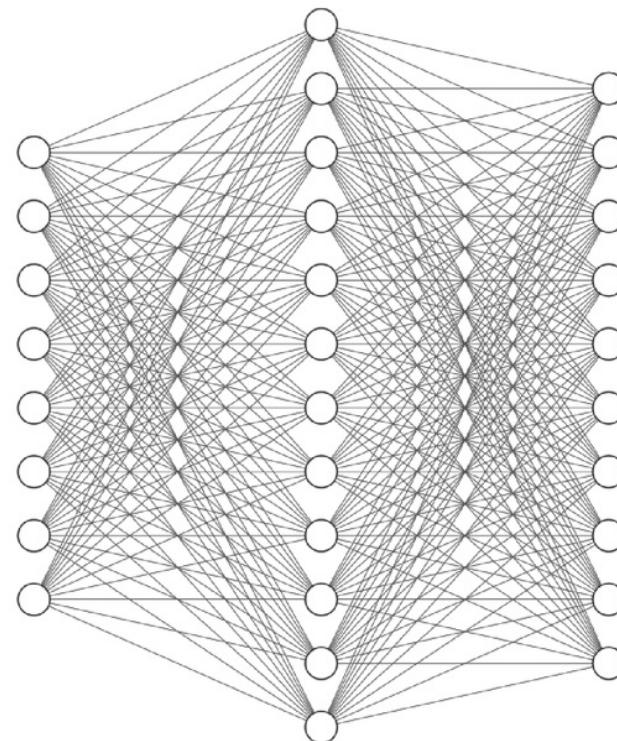
ดังนั้นหากนำชุดข้อมูลมาต่อ กัน 20 ตัว จะกลายเป็นชุดข้อมูล 1 ชุด และในการฝึก ไม่เดล จะใช้ชุดข้อมูลทั้งหมดอย่างน้อย 50 ตัว

3.6 การออกแบบโมเดล Machine Learning

ไม่เดลที่ใช้ในการฝึกทำท่าทางที่มีข้อมูลของเซนเซอร์ในแต่ละนิ้ว เดือกเป็นโครงสร้างค่ายประสาท เทียม (Neural Network) เพราะมีชุดข้อมูลเยอะในการฝึกทำท่าทาง โครงสร้างประสาทเทียมช่วยทำให้ ไม่ต้องหาคุณลักษณะ (Feature) ให้กับข้อมูล และใช้การเรียนแบบมีการสอน (Supervised Learning) ที่คือการเรียนที่มีการตรวจสอบเพื่อให้โครงสร้างประสาทเทียมปรับตัวเอง เพื่อให้ได้ คำตอบที่ดีขึ้น โดยไม่เดลสำหรับข้อมูลที่ใช้ในการฝึกจะทดสอบทั้งหมด 3 ไม่เดลดังนี้²⁶

3.6.1 การออกแบบโมเดลของอุปกรณ์แบบที่ 1 สำหรับท่านิ้ง

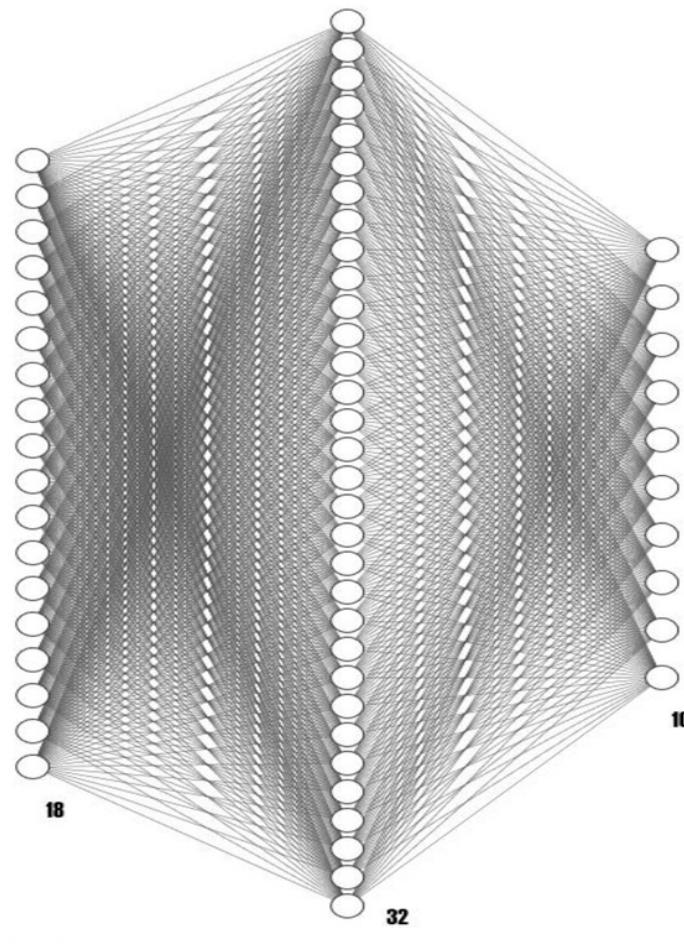
เป็นไม่เดลโครงสร้างประสาทเทียมธรรมชาติ (Neural Network) จะใช้ Library ของ Scikit-learn ในการสร้างไม่เดล มีคุณลักษณะ (Feature) ทั้งหมด 8 ลักษณะ ประกอบด้วยเซนเซอร์ วัดความอุตุนิวทั้งหมด 5 ลักษณะ เซนเซอร์วัดมุมข้อมือ 3 ลักษณะ เพื่อให้ได้ผลลัพธ์ เป็น ท่าทางของมือรูปแบบนึง ก่อนที่จะนำคุณลักษณะทั้งหมดไปฝึกจำเป็นต้องแปลง (Transform) ก่อน เพื่อให้การทำงานของผลลัพธ์แม่นยำยิ่งขึ้น โดยโครงสร้างของโครงสร้างประสาทเทียมประกอบด้วย ชั้น ขาเข้า (Input layer) คือลักษณะทั้งหมด 8 ชั้น ชั้นซ่อน (Hidden layer) ใช้เป็น 12 ชั้น



รูป 3.11 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 1

3.6.2 การออกแบบโมเดลของอุปกรณ์แบบที่ 2 สำหรับท่านิ่ง

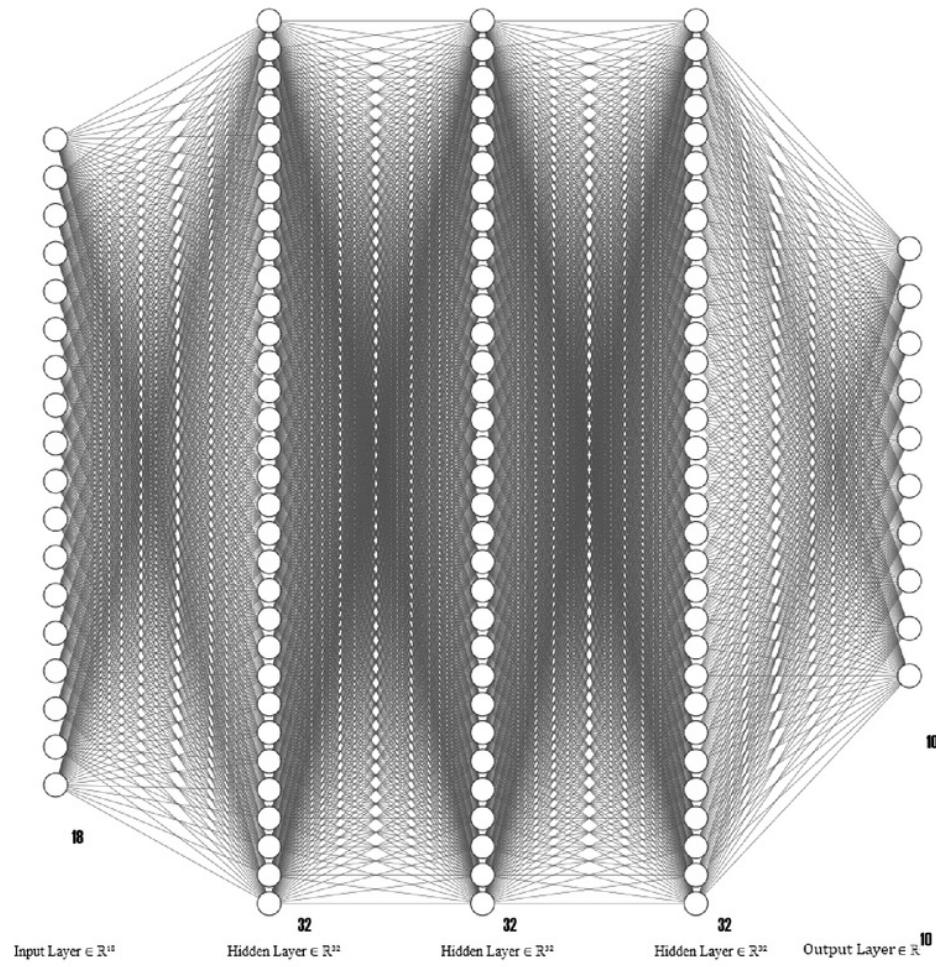
เป็นโมเดลโครงข่ายประสาทเทียมธรรมชาติ ใช้ Library ของ Keras ซึ่งมี Tensorflow เป็น Back-end ในการสร้าง โมเดล มีคุณลักษณะ (Feature) ทั้งหมด 18 คุณลักษณะ ประกอบด้วย特征เซอร์วัคัมุมของแต่ละนิ่วทั้งหมด 15 คุณลักษณะ เซอร์วัคัมุมข้อมูล 3 คุณลักษณะ เพื่อให้ได้ผลลัพธ์เป็นทางของมือรูปแบบนิ่ง โครงสร้างของประสาทเทียมประกอบด้วย ชั้นขาเข้าคือคุณลักษณะทั้งหมด 18 ชั้น และชั้นซ่อน ใช้เป็น 32 ชั้น ส่วนชั้นขาออก (Output layer) ใช้ตามจำนวนของท่าทั้งหมดที่ฝึกเป็น 10 ชั้น



รูป 3.12 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 2

3.6.3 การออกแบบโมเดลของอุปกรณ์แบบที่ 2 สำหรับทำนิ่งและทำคลื่นไฟฟ้า

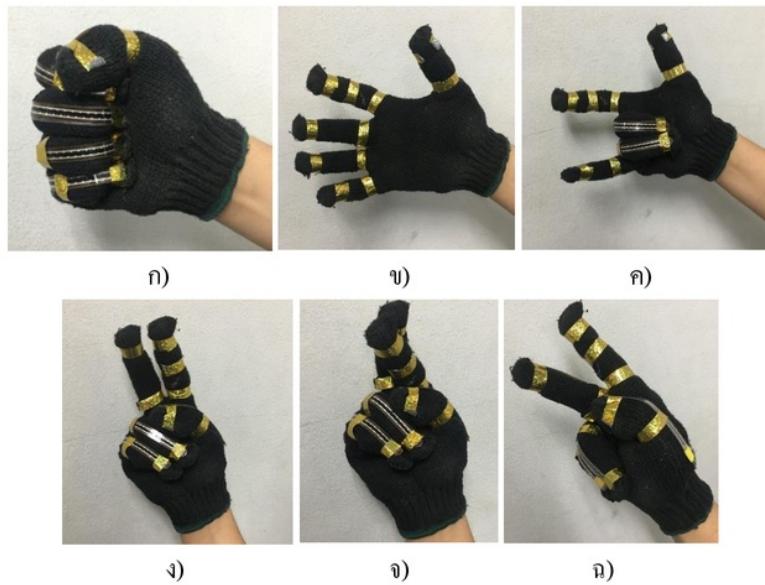
เป็นโมเดลโครงข่ายประสาทเทียมที่ออกแบบมาสำหรับประมวลผลลำดับ (LSTM) จะใช้ Library ของ Keras ในการสร้างโมเดล มีคุณลักษณะทั้งหมด 18 ลักษณะ ประกอบด้วย เช่นเชอร์วัคคุณของแต่ละนาทีทั้งหมด 15 ลักษณะ เช่นเชอร์วัคคุณข้อมือ 3 ลักษณะ โดยมีลำดับของ ทำทางทั้งหมด 20 ครั้ง (Timestep) เพื่อให้ได้ผลลัพธ์เป็นทำทางของมือทั้งทำนิ่งและคลื่นไฟฟ้า โครงสร้างของประสาทเทียมประกอบด้วย ชั้นขาเข้าเป็นอาร์กัมณะ ($20, 18$) 20 คือจำนวนครั้ง ของลำดับ ส่วน 18 คือคุณลักษณะทั้งหมด ชั้นซ่อนใช้เป็น 32 ชั้น ทั้งหมด 3 เลเยอร์ และชั้นขา ออกใช้เป็นจำนวนทำทั้งหมดในการฝึก มีทั้งหมด 10 ชั้น



รูป 3.13 สถาปัตยกรรมโครงข่ายประสาทเทียมโมเดลที่ 3

บทที่ 4 การทดลอง

การทดลองในเบื้องต้น คือการทดลองเพื่อค้นหาวิธีในการใช้ถุงมือ และวิธีการตรวจจับท่าทางมือให้ได้แม่นยำที่สุด โดยท่าทางที่เลือกมา มีทั้งสิ้น 9 ท่าทาง ที่มีลักษณะแตกต่างกัน 9 แบบ ซึ่งมีความสำคัญในการวัดความความแม่นยำในการส่วนต่าง ๆ ของถุงมือ ท่าทางที่เลือกมาใช้ในการทดลองมีดังนี้



รูป 4.1 ท่าที่ใช้ในการทดสอบ

- ก) กำมือ
- ข) แบบมือ
- ค) บอกรัก
- ง) ชูสองนิ้ว
- จ) ไขวนิ้ว
- ฉ) พยัญชนะ ก



(ก) (ข)

รูป 4.2 ท่าเคลื่อนไหวที่ใช้ในการทดสอบ - พยัญชนะ ข

- (ก) ขั้นตอนแรก
- (ข) ขั้นตอนที่สอง



(ก) (ข) (ค)

รูป 4.3 ท่าเคลื่อนไหว - พยัญชนะ ค

- (ก) ขั้นตอนแรก
- (ข) ขั้นตอนที่สอง
- (ค) ขั้นตอนที่สาม



(ก) (ข)

รูป 4.4 ท่าเคลื่อนไหว - พยัญชนะ ง

- (ก) ขั้นตอนแรก
- (ข) ขั้นตอนที่สอง

ทำก้ามือและแบบมือ เป็นการทำพื้นฐานที่ถูกมือตรวจจับทำทางเครื่องสามารถตรวจสอบได้ ทำบอกรัก เป็นการทำที่ใช้ทดสอบกรณีค่าของทุกนิ่วไม่ทำกัน ทำชูสองนิ่วและไขว้นิ่ว เป็นการทำที่ทดสอบการแยกและทำในกรณีที่ทำทั้งสองแตกต่างกันในแกนอื่นที่ไม่ใช่แกนหลักของการงอนนิ่ว และทำพยัญชนะเป็นการทำที่ทดสอบการตรวจจับทำที่มีการเคลื่อนไหว

การทดลองแบ่งได้เป็น 4 ขั้นตอนหลัก ๆ ได้แก่ การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ การทดลองเกี่ยวกับเว็บแอพพลิเคชัน และการทดลองเกี่ยวกับ Machine Learning

4.1 การทดลองการเก็บค่าจากถุงมือทั้งสองแบบ

คือการทดลองเกี่ยวกับถุงมือที่ได้ออกแบบมาทั้งสองแบบ ว่าสามารถจับค่าได้แม่นยำและมีปัญหาในการจับค่าหรือไม่

4.1.1 การทดลองใช้ถุงมือแบบที่ 1

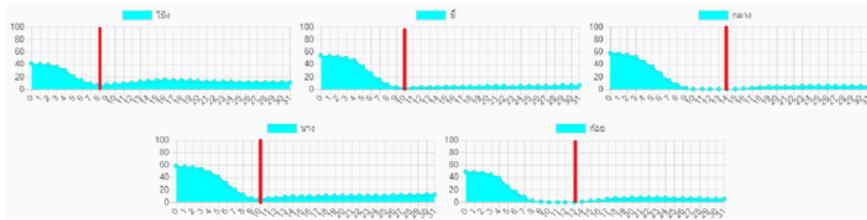
4.1.1.1 วิธีการทดลอง

ประดิษฐ์ถุงมือแบบที่ 1 ขึ้นตามที่ออกแบบไว้ จากนั้นลองอ่านข้อมูลที่ได้จากเซนเซอร์ทุกด้าน (เซนเซอร์วัดค่าความงอ 5 ตัว และเซนเซอร์หน่วยวัดศาสัยหลักความเลื่อย 1 ตัว) สำหรับเซนเซอร์วัดค่าความงอจะมีค่าตั้งแต่ 0-1023 และเซนเซอร์หน่วยวัดศาสัยหลักความเลื่อยจะอ่านเพียงค่าเซนเซอร์หน่วยวัดศาสัยหลักความเลื่อยอย่างเดียว 3 แกนและเซนเซอร์หน่วยวัดความเร็ว เชิงมุม 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 จากนั้นลองทำท่าที่นิ่งเมื่องดัน ได้แก่ทำก้ามือ แบบมือชูนิ่วแต่ละนิ่ว ชูสองนิ่ว และไขว้นิ่ว

4.1.1.2 ผลการทดลอง

ถุงมือไม่มีปัญหาด้านการจับค่า สามารถทำการจับค่าได้ตามปกติ ยกเว้นในบางกรณีที่จับค่าไม่ได้หรือเพี้ยน ดังนี้

- 1) กรณีแบบมือจ่อนสุด จะทำให้เซนเซอร์วัดค่าความงอเกิดการงอ ถุงมือจะจับค่าได้เท่ากับการงอนิ่ว จากรูป 4.6 สามารถสังเกตได้ว่า หลังจากเส้นสีแดงค่ามีแนวโน้มเพิ่มขึ้นเล็กน้อย เกิดจากการงอของเซนเซอร์วัดค่าความงอ เมื่อแบบมือมากเกินไป ในผู้ใช้งานรายจางแบบมือได้มากกว่านี้ ซึ่งอาจจะทำให้ไม่สามารถแยกท่าบ้างทำอออกกันได้



รูป 4.5 แผนภูมิข้อมูลของท่าแบบมือ (ถุงมือแบบที่ 1)

- 2) กรณีที่มีการไขว้น้ำ ค่าจะอยู่น้อยกว่าเกณฑ์ เช่นเชอร์วัคค่าความคงสามารถ
จับได้ ตัวอย่างเช่นการไขว้น้ำที่กันน้ำ梧桐 ค่าจะได้ใกล้เคียงกับการงาน
น้ำทั้งสองปกติ
- 3.) กรณีที่งอน้ำปะปังไปหาลายทิศทาง ค่าจะอยู่น้อยกว่าเกณฑ์ เช่นเชอร์วัคค่าความ
คงสามารถจับได้ช่วนกัน ตัวอย่างเช่น มุนระหวน้ำปะปังกันน้ำที่ การงาน
เป็นมุมจากกับการแนวบริเวณน้ำทั้งสองจะได้ค่าเหมือนกัน

4.1.2 การทดลองใช้ถุงมือแบบที่ 2

4.1.2.1 วิธีการทดลอง

ประดิษฐ์ถุงมือแบบที่ 2 ขึ้นตามที่ออกแบบไว้ จากนั้นลองอ่านข้อมูลที่ได้จาก
เช่นเชอร์ทุกด้าน (เช่นเชอร์หน่วยวัดอัศัยหลักความเร็วอย่าง 6 ตัว) สำหรับเช่นเชอร์หน่วยวัดอัศัยหลัก
ความเร็วอย่างทุกด้าน จะอ่านเพียงค่าเช่นเชอร์หน่วยวัดอัศัยหลักความเร็วเชิงเส้น 3 แกนและ
เช่นเชอร์หน่วยวัดความเร็วเชิงมุม 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 จากนั้นลองทำท่าที่
เมื่องดัน ได้แก่ท่ากำมือ แบบมือ ชูน้ำแต่ละน้ำ ชูสองน้ำ และไขว้น้ำ

4.1.1.2 ผลการทดลอง

ถุงมือไม่มีปัญหาด้านการจับค่า ท่าที่เคยมีปัญหาในถุงมือแบบที่ 1 มีค่า
แตกต่างกันอย่างชัดเจน ทำให้สามารถแยกท่าที่เคยมีปัญหาออกจากกันได้

จากการทดลอง 4.1 ทั้งหมดพบว่า ถุงมือแบบที่ 2 มีโอกาสในการจับค่าผิดพลาดที่น้อยกว่า แต่
ขังต้องนำค่าไปปรับปรุงและประยุกต์ใช้ต่อไป

4.2 การทดลองการปรับปรุงและประยุกต์ค่าจากถุงมือ

เนื่องจากค่าที่ได้จากเช่นเชอร์หน่วยวัดอัศัยหลักความเร็วอย่างโดยตรงนั้นเป็นค่าที่มีช่วงกว้าง
และไม่สามารถนำไปใช้ในขั้นตอนการทดลองเกี่ยวกับ Machine Learning ได้โดยตรง การทดลองนี้
จึงเป็นการนำค่าโดยตรงจากถุงมือมาปรับปรุงหรือประยุกต์ใช้ในการหาค่าใหม่ที่มีประโยชน์มาก
ขึ้นก่อนนำมาวิเคราะห์ด้วย Machine Learning โดยใช้วิธีต่างๆ

4.2.1 การทดลองใช้สูตรคำนวณทามุนจากเซนเซอร์วัดความเร่ง

4.2.1.1 วิธีการทดลอง

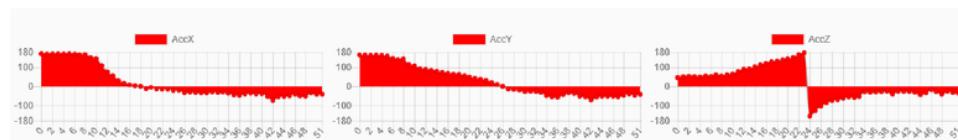
7

อ่านค่าของเซนเซอร์วัดความเร่งชิงสั้น 3 แกน แล้วนำมาคำนวณตามสูตร

คำนวณทามุนจากเซนเซอร์วัดความเร่ง ให้กลไกเป็นค่ามุน 3 ค่า โดยเซนเซอร์หน่วยวัดอัศยหลัก ความเฉื่อยที่อยู่หลังมือจะมีค่ามุนระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอัศยหลักความเฉื่อยที่อยู่ปลายนิ้วจะนำไปหาความแตกต่างจาก เซนเซอร์หน่วยวัดอัศยหลักความเฉื่อยที่อยู่หลังมือก่อน กลไกเป็นค่าความแตกต่างระหว่างช่วง -180 ถึง 180 จากนั้นลองทำท่าท่านั่งเบื้องต้น ขับมือชา และ ขับมือเร็ว

4.2.1.2 ผลการทดลอง

สำหรับท่านนี้จะทำการขับอย่างชา ๆ จะได้ค่ามุน 3 แกนที่แม่นยำ แต่เมื่อมีการ ขับมืออย่างรวดเร็ว จะทำให้มีค่ามุนที่เพียงเล็กน้อย เกิดจากธรรมชาติของเซนเซอร์วัดความเร่ง



รูป 4.6 แผนภาพชุดข้อมูลเฉพาะนิรบุรุษของท่านนี้ที่แบนมือ (ลุงมีอแบบที่ 2)

4.2.2 การทดลองใช้การคำนวณมุนด้วยตัวกรองร่วมสองแบบ

4.2.2.1 วิธีการทดลอง

อ่านค่าของเซนเซอร์วัดความเร่งชิงสั้น 3 แกน แล้วนำมาคำนวณตามขั้นตอน ของตัวกรองร่วมสองแบบ ให้กลไกเป็นค่ามุน 3 ค่า โดยเซนเซอร์หน่วยวัดอัศยหลักความเฉื่อยที่อยู่หลังมือจะมีค่ามุนระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอัศยหลักความเฉื่อยที่อยู่ปลายนิ้วจะนำไปหาความแตกต่างจาก เซนเซอร์หน่วยวัดอัศยหลักความเฉื่อยที่อยู่หลังมือก่อน กลไกเป็นค่าความแตกต่างระหว่าง -180 ถึง 180 จากนั้นลองทำท่าท่านั่งเบื้องต้น ขับมือชา ขับมือเร็ว และ ลองปรับค่า α ในสูตรระหว่างช่วง (0,1) ให้คำนวณค่าที่ผิดพลาดน้อยที่สุด

4.2.2.2 ผลการทดลอง

ที่ค่า $\alpha=0.6$ จะได้ค่ามุน 3 มุนที่แม่นยำ และไม่มีปัญหาเมื่อทำการขับมืออย่างรวดเร็ว แต่เมื่อมีการขับมือเกินค่า 360 องศา หรือต่ำกว่า 0 องศา การคำนวณด้วยตัวกรองร่วมสองแบบจะพยายามทำให้ค่าเปลี่ยนไปอย่างชา ๆ ซึ่งทำให้ค่าเพียงไปจากความเป็นจริงพอสมควร

4.2.3 การทดลองใช้การคำนวณมุมด้วยตัวกรองค่าความ

4.2.3.1 วิธีการทดลอง

อ่านค่าของเซนเซอร์วัดความเร่งชิงสีน 3 แกน แล้วนำมาคำนวณตามขั้นตอน ของตัวกรองค่าความ ให้กลไกเป็นค่ามุม 3 ค่า โดยเซนเซอร์หน่วยวัดอศาสัยหลักความเฉื่อยที่อยู่หลัง มือจะมีค่ามุมระหว่าง 0 ถึง 360 และเซนเซอร์หน่วยวัดอศาสัยหลักความเฉื่อยที่อยู่ปลายนิ้วจะนำໄไป หาความแตกต่างจาก เซนเซอร์หน่วยวัดอศาสัยหลักความเฉื่อยที่อยู่หลังมือก่อน กลไกเป็นค่าความ แตกต่างระหว่างช่วง -180 ถึง 180 จากนั้นลองทำท่าทันนึงเมื่องดัน ขยับมือซ้าย และขยับมือขวา

4.2.3.2 ผลการทดลอง

ได้ค่ามุม 3 มุมที่แม่นยำ และไม่มีปัญหามือทำการขยับมืออย่างรวดเร็ว เมื่อมี ค่ามุมเกินค่า 360 องศา หรือต่ำกว่า 0 องศา การคำนวณด้วยตัวกรองค่าความจะยังคงสร้างค่าที่เพียง ไปจากความเป็นจริงเหมือนกับการใช้ตัวกรองร่วมสองแบบ และการทำงานของ ไมโครคอนโทรลเลอร์โดยรวมชัดเจน และมีแนวโน้มที่จะใช้หน่วยความจำขนาดใหญ่ได้เร็วขึ้น

4.2.4 การทดลองการตรวจจับการเคลื่อนไหว

4.2.5.1 วิธีการทดลอง

กำหนดค่าขั้นต่ำของการเคลื่อนไหว (Threshold) ของมุมต่าง ๆ ไว้ และให้ บันทึกข้อมูลในเฉพาะข้อมูลที่มีการเคลื่อนไหวจากจุดล่าสุดมากกว่าค่าขั้นต่ำท่านั้น ทดลองปรับค่า ขั้นต่ำภายในช่วง 5-50 องศา

4.2.5.2 ผลการทดลอง

ข้อมูลที่บันทึกเมื่อทำท่าทันนึงมีค่าที่คงที่มากขึ้น และสามารถเริ่มตรวจจับการ เคลื่อนไหวเพื่อเปลี่ยนท่าได้ที่สุดเมื่อค่าขั้นต่ำมีค่า 30 องศา

จากการทดลอง 4.2 ทั้งหมดพบว่า วิธีการในการคำนวณมุมที่เหมาะสมที่สุดในเบรกเกอร์และ ประสิทธิภาพคือการใช้ตัวกรองร่วมสองแบบ และเริ่มมีการตรวจจับการเคลื่อนไหวเมื่อมีการ เปลี่ยนแปลงมุมขั้นต่ำ 30 องศา ในแกนใดแกนหนึ่งของเซนเซอร์ทั้งหมด

4.3 การทดลองเกี่ยวกับเว็บแอปพลิเคชัน

คือการทดลองพัฒนาเว็บแอปพลิเคชันในส่วนต่าง ๆ รวมถึงการเชื่อมต่อ แลกเปลี่ยนข้อมูลกับเซิร์ฟเวอร์ เพื่อนำข้อมูลจากเซิร์ฟเวอร์มาประยุกต์ใช้งานเกี่ยวกับ Machine Learning หรือขั้นตอนอื่น ๆ ของโครงการต่อไป

4.3.1 การทดลองการส่งค่าผ่าน Socket.IO

4.3.1.1 วิธีการทดลอง

เชื่อมต่อเว็บแอปพลิเคชันเข้ากับเซิร์ฟเวอร์ผ่านทาง Socket.IO เมื่อเริ่มเข้าเว็บไซต์ สังขุดข้อมูลจำนวน 20 ชุดต่อวินาทีจากเซิร์ฟเวอร์ไปยังเว็บแอปพลิเคชัน จากนั้นลองกดปุ่มนบบันเว็บแอปพลิเคชันเพื่อส่งข้อมูลค่าสุ่มทั้ง 20 ชุดกลับไปวิเคราะห์ที่เซิร์ฟเวอร์

4.3.1.2 ผลการทดลอง

ทั้งสองฝ่ายสามารถส่งข้อมูลหากันได้ทุกครั้งตลอดกระบวนการ

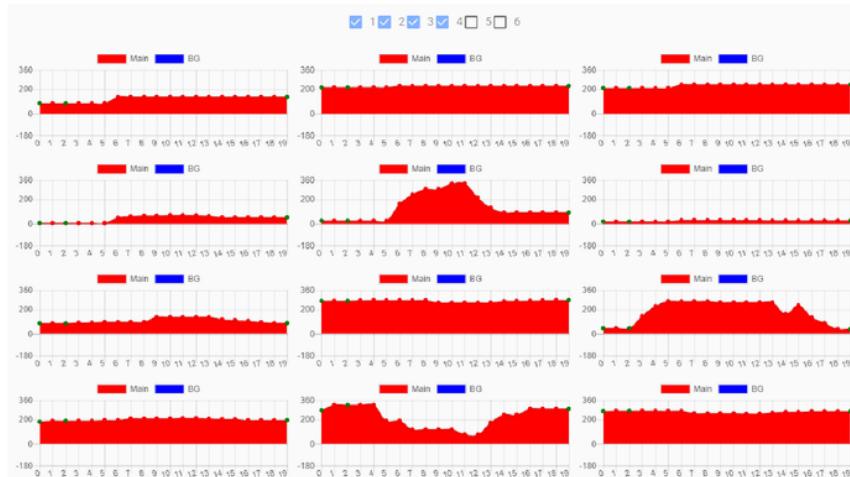
4.3.2 การทดลองการแสดงข้อมูลแบบเปลี่ยนแปลงตามเวลา

4.3.2.1 วิธีการทดลอง

เชื่อมต่อเว็บแอปพลิเคชันเข้ากับเซิร์ฟเวอร์ผ่านทาง Socket.IO เซิร์ฟเวอร์ส่งค่าอย่างต่อเนื่องมาให้เว็บแอปพลิเคชัน สร้างแผนภูมิเส้นแสดงค่า 20 ค่าล่าสุดจากเซ็นเซอร์แต่ละค่าที่ส่งมาจากเซิร์ฟเวอร์

4.3.2.2 ผลการทดลอง

สามารถแสดงค่าได้ตรงตามที่อ่านได้จากเซิร์ฟเวอร์และเปลี่ยนแปลงตามเวลาโดยมีความล่าช้าน้อยกว่าหนึ่งวินาที



รูป 4.7 การแสดงผลข้อมูลเป็นแผนภูมิเส้นที่เปลี่ยนแปลงตามเวลา

4.3.3 การทดลองบันทึกชุดข้อมูลของท่าต่าง ๆ และส่งออกเป็นไฟล์ .CSV

4.3.3.1 วิธีการทดลอง

ในหน้าจอพัฒนา สำหรับการบันทึกรายครั้ง ตั้งค่าชื่อท่าแล้วกดปุ่ม RECORD เพื่อเริ่มนับที่ก้าวค่าที่ส่งมาจากเซิร์ฟเวอร์ ระหว่างนับที่ก้าวทำตามที่ต้องการ กดปุ่ม Stop เพื่อหยุดบันทึก จากนั้นกดปุ่ม Save เพื่อบันทึกท่า ทำแบบนี้จนครบ 50 ครั้ง เพื่อให้ได้ชุดข้อมูลจำนวนที่ขึ้นต่อสำหรับหนึ่งท่า

สำหรับการบันทึกต่อเนื่อง กดปุ่ม AUTO เพื่อเริ่มนับที่ก้าวค่าที่ส่งมาจากเซิร์ฟเวอร์ ทุก ๆ 20 ค่า ค่าถัดไปจะถูกตัดเป็นชุดข้อมูลใหม่ ระหว่างนับที่ก้าวทำตามที่ต้องการทำจนครบอย่างน้อย 50 ครั้ง กดปุ่ม Stop เพื่อยุดบันทึก จากนั้นกดปุ่ม Save เพื่อบันทึกท่า เพื่อให้ได้ชุดข้อมูลจำนวนที่ขึ้นต่อสำหรับหนึ่งท่า

จากนั้นกดปุ่ม Export เพื่อส่งออกชุดข้อมูลทั้งหมดออกเป็นไฟล์ export.csv

4.3.3.2 ผลการทดลอง

ทั้งการบันทึกรายครั้งและบันทึกต่อเนื่อง สามารถบันทึกค่าลงเป็นไฟล์ csv และนำไปใช้ในการกระบวนการฝึกโมเดล Machine Learning ได้อย่างไม่มีปัญหา

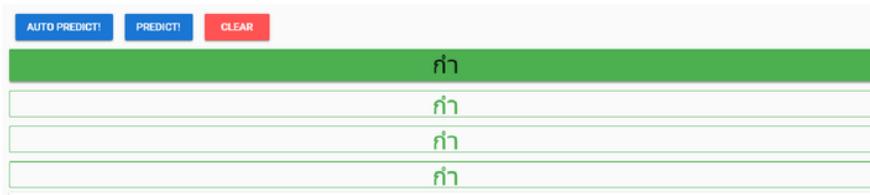
4.3.4 การทดลองส่งข้อมูลล่าสุดไปตรวจสอบท่าแล้วแสดงผลตอบกลับ

4.3.4.1 วิธีการทดลอง

ในหน้าจอพัฒนา กดปุ่ม PREDICT เพื่อส่งข้อมูล 20 ค่าล่าสุดไปยังเซิร์ฟเวอร์ผ่านทาง Socket.IO ทางฝ่ายเซิร์ฟเวอร์จำลองการส่งค่าตอบกลับไปยังเว็บแอพพลิเคชั่น แสดงผลตอบกลับที่ได้รับ

4.3.4.2 ผลการทดลอง

สามารถจำลองการส่งข้อมูลไปวิเคราะห์และแสดงผลตอบกลับได้ พร้อมที่จะใช้งานเพื่อแสดงผลคำตอบจริงในการทดลองถัดไป



รูป 4.8 การจำลองการแสดงผลการตรวจสอบท่าทาง

4.4 การทดลองเกี่ยวกับ Machine Learning

คือการนำข้อมูลที่ผ่านการปรับปรุงแล้วมาทดลองเกี่ยวกับโมเดล Machine Learning ต่าง ๆ ที่ใช้ในการวิเคราะห์ท่าทางมือ โดยการทดลองอัลกอริธึมการสร้างโมเดลที่แตกต่างกัน หรือการใช้ค่าที่แตกต่างกัน

4.4.1 การทดลองโมเดลแบบที่ 1: การฝึกโมเดล

4.4.1.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการสำรวจของเว็บแอพพลิเคชั่น แบ่งข้อมูลที่จะนำเข้าไปฝึกโดยใช้ข้อมูลร้อยละ 70 ของทั้งหมดในการฝึก ส่วนที่เหลือใช้ในการประเมินผลลัพธ์ของโมเดล (Evaluate) นำข้อมูลที่จะฝึกไปเปลี่ยนรูป (Transform) ให้ข้อมูลอยู่ในช่วงเดียวกัน หลังจากนั้นสร้างโมเดลขึ้นมา โดยใช้จำนวนครั้งในการฝึก (Epochs) 10,000 ครั้ง จึงนำโมเดลไปประเมินด้วยข้อมูลที่ไม่ได้นำไปฝึก

ถูกเมื่อที่ใช้ในโมเดลนี้คือถูกเมื่อบนแบบที่ 1 ท่าที่ใช้ในการฝึกและทดสอบโมเดลนี้ได้แก่ กำมือ, แบบมือ, บอกรัก, ชูสองนิ้ว และไขว่นิ้ว

4.4.1.2 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูงมาก ผลจากการประเมินสูงเกือบถึงร้อยละ 100 แต่ไม่สามารถทำนายท่าทางเคลื่อนไหวได้

4.4.2 การทดลองโมเดลแบบที่ 1: ทดสอบความแม่นยำ

4.4.2.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัพโหลดลงไปบนไซต์โคดคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำการทดสอบ กดปุ่มผ่านหน้าเว็บแอพพลิเคชั่นเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำท่าที่ทั้งหมด 50 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

4.4.2.2 ผลการทดลอง

จากการนำไปทำนายท่าที่นี่แต่ละท่า ทั้งหมด 50 ครั้ง ถูกต้อง 41 ครั้งหรือค่าความแม่นยำร้อยละ 82 เนื่องจากมีบางท่าที่เป็นข้อจำกัดของถูกเมื่อบนแบบที่ 1 ทำให้การทำนายเกิดข้อผิดพลาด

ตาราง 4.1 ผลการทดสอบความแม่นยำโมเดลแบบที่ 1

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบบมือ	10	10	100%
บอกรัก	10	10	100%
ชูสองนิ้ว	10	10	100%
ไขว้นิ้ว	10	1	10%
รวม	50	41	82%

ตาราง 4.2 Confusion Matrix ของโมเดลแบบที่ 1

		ท่าจริง					Precision
		กำมือ	แบบมือ	บอกรัก	ชูสองนิ้ว	ไขว้นิ้ว	
ท่าที่ตรวจสอบ	กำมือ	10	0	0	0	0	1
	แบบมือ	0	10	0	0	0	1
	บอกรัก	0	0	10	0	0	1
	ชูสองนิ้ว	0	0	0	10	9	0.53
	ไขว้นิ้ว	1	0	0	0	1	1

4.4.3 การทดลองโมเดลแบบที่ 2: การฝึกโมเดล

4.4.3.1 วิธีการทดลอง

อ่านค่าข้อมูล Dataset ที่ได้จากการอัดส่วนของเว็บแอพพลิเคชัน แบ่งข้อมูลที่นำเข้าไปฝึกเป็นร้อยละ 70 นำข้อมูลไปเปลี่ยนรูป หลังจากนั้นสร้างโมเดลขึ้นมาโดยเบลี่ยนชั้นซ่อน (Hidden layer) เรื่อยๆ จนเจอกำาหนดสมบูรณ์มากขึ้น ปรับเปลี่ยนโมเดลไม่ให้ค่าฟิตเกินไปโดยใช้ Dropout ใช้จำนวนครั้งในการฝึกประมาณ 150 ครั้ง จึงนำโมเดลไปประเมิน

ถุงมือที่ใช้ในโมเดลนี้คือถุงมือแบบที่ 2 ที่ใช้ในการฝึกและทดสอบโมเดลนี้ได้แก่ กำมือ, แบบมือ, บอกรัก, ชูสองนิ้ว และไขว้นิ้ว

4.4.3.2 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำสูง ผลจากการประเมินสูงเกินร้อยละ 100 เช่นเดียวกับโมเดลแบบแรก ไม่สามารถทำนายท่าคลื่อนไหวได้เช่นเดียวกัน

4.4.4 การทดสอบโมเดลแบบที่ 2: ทดสอบความแม่นยำ

4.4.4.1 วิธีการทดสอบ

นำโมเดลที่ได้จากขั้นตอนการฝึกอัพโหลดลงไปบนไมโครคอนโทรลเลอร์เพื่อเรียกใช้โดยเซิร์ฟเวอร์ ทำการทดสอบ กดปุ่มผ่านหน้าเว็บแอพพลิเคชันเพื่อเรียกให้เซิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำท่าที่ง่ายทั้งหมด 50 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

4.4.4.2 ผลการทดสอบ

จากการนำไปทำนายท่าที่ง่ายทั้งหมด 50 ครั้ง ถูกต้อง 48 ครั้ง หรือค่าความแม่นยำร้อยละ 96 เนื่องจากการใช้คุณลักษณะที่ 2 ลดข้อจำกัดด้านการรองน้ำลง ทำให้ผลลัพธ์ที่ได้แม่นยามากขึ้น

ตาราง 4.3 ผลการทดสอบความแม่นยำโมเดลแบบที่ 2

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบบมือ	10	10	100%
บอกรัก	10	10	100%
ชูสองนิ้ว	10	10	100%
ไขว้หัน	10	8	80%
รวม	50	48	96%

ตาราง 4.4 Confusion Matrix ของโมเดลแบบที่ 2

		ท่าจริง					Precision
		กำมือ	แบบมือ	บอกรัก	ชูสองนิ้ว	ไขว้หัน	
ท่าตรวจสอบ	กำมือ	10	0	0	0	0	1
	แบบมือ	0	10	0	0	0	1
	บอกรัก	0	0	10	0	0	1
	ชูสองนิ้ว	0	0	0	10	2	0.83
	ไขว้หัน	0	0	0	0	8	1

4.4.5 การทดลองโมเดลแบบที่ 3: การฝึกโมเดล

4.4.5.1 วิธีการทดลอง

อ่านค่าชุดข้อมูล (Dataset) ที่ได้จากการอัดส่วนของเว็บแอพพลิเคชั่น ทำการแบ่งกลุ่มของข้อมูลเป็นคำนับของเวลา คือ ข้อมูลของมุมทั้งหมดในช่วงเวลาที่ 1 (เริ่มต้นท่า) ไปจนถึงข้อมูลของช่วงเวลาที่ 20 (สิ้นสุดท่า) นำข้อมูลทั้ง 20 ช่วงเวลา ไปฝึกฝนเป็นท่าหนึ่งท่า ใช้ Dropout ในการปรับไม่ให้มีผลต่อการฝึก เนื่องจากในช่วงเวลาที่ 1 จานวนครั้งในการฝึกประมาณ 100 ครั้ง

ถูกเมื่อที่ใช้ในโมเดลนี้คือถูกเมื่อบนที่ 2 ท่าที่ใช้ในการฝึกและทดสอบ โมเดลนี้ได้แก่ กำเมือง, แบบเมือง, ชุมชนน้ำ, ไขว่น้ำ, พืชผักชุมชน ก, ข, ค และ ง

4.4.5.1 ผลการทดลอง

โมเดลที่ได้มีความแม่นยำ แต่เนื่องจากเป็น โมเดลที่ประมวลผลคำนับด้วยทำให้โมเดลมีความซับซ้อน ใช้เวลาในการฝึกค่อนข้างนานและจำเป็นต้องใช้ชุดข้อมูลจำนวนมาก เพื่อให้ได้ผลลัพธ์ในทุกๆกรณีที่จะเกิด

4.4.6 การทดลองโมเดลแบบที่ 3: ทดสอบความแม่นยำ

4.4.6.1 วิธีการทดลอง

นำโมเดลที่ได้จากขั้นตอนการฝึกอัพโหลดลงไปบนไซต์โคร์โคน โගรลเลอร์ เพื่อเรียกใช้โดยเชิร์ฟเวอร์ ทำการทดสอบ กดปุ่มผ่านหน้าเว็บแอพพลิเคชั่นเพื่อเรียกให้ เชิร์ฟเวอร์ทดสอบความแม่นยำแล้วแสดงผลคำตอบ โดยการลองทำท่าหนึ่งแล้วคลิกท่องไปทั้งหมด 80 ครั้ง ว่าผลลัพธ์ถูกต้องตามที่ต้องการหรือไม่

4.4.6.2 ผลการทดลอง

จากการนำไปทำนายท่าหนึ่งและทำคลิกท่องไปทั้งหมด 90 ครั้ง ถูกต้อง 70 ครั้ง หรือความแม่นยำร้อยละ 77.77 เนื่องจากมีการทำคลิกท่องไปเพิ่มมา และทำท่าหนึ่งที่ ใกล้เคียงทำกันมากขึ้น ทำให้เกิดข้อผิดพลาดมากที่สุดใน 3 โมเดลที่ผ่านมา โดยเฉพาะกลุ่มท่า ชุมชนน้ำ, ไขว่น้ำ และพืชผักชุมชน ก

ตาราง 4.5 ผลการทดสอบความแม่นยำโมเดลแบบที่ 3

ชื่อท่า	จำนวนครั้งที่ทดสอบ	จำนวนครั้งที่ถูก	ความแม่นยำ
กำมือ	10	10	100%
แบบมือ	10	10	100%
บลอกรักก	10	10	100%
ชูสองนิ้ว	10	9	90%
ไขวนิ้ว	10	4	40%
ก	10	5	50%
ข	10	8	80%
ค	10	6	60%
จ	10	8	80%
รวม	90	70	77.77%

ตาราง 4.6 Confusion Matrix ของโมเดลแบบที่ 3

បររបាណក្រម

- [1] Felipe, A. Q. Marcelo, R. Rafael, R. D. Enzo, H. W and Alessandro, G. “**A Gesture Detection Glove for Human-Computer Interaction.**” Federal University of Pampa, Alegrete - RS, Brazil
- [5] [2] Rung-Huei, L. Ming, O. “**A real-time continuous gesture recognition system for sign language.**” Dept. of Inf. Manage., Shieh Chen Univ., Taipei, Taiwan
- [12] [3] Ji-Hwan, K. Nguyen, D. T. Tae-Seong, K. “**3-D hand motion tracking and gesture recognition using a data glove.**” Department of Biomedical Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, 446-701 Gyeonggi-do, Korea
- [4] FriendlyARM. 2018. **NanoPI-DUO Hardware spec.** [Online] Available: http://wiki.friendlyarm.com/wiki/index.php/NanoPi_Duo#Hardware_Spec
- [18] [5] Polulu. 2016. **MinIMU-9 v5 Gyro, Accelerometer, and Compass.** [Online] Available: <https://www.pololu.com/product/2738>
- [6] Doratong24. 2017. **LSTM.** [Online] Available: <https://medium.com/@tongkornkitt/a3a55cd37883>
- [14] [7] Arun Gupta. 2017. **REST vs WebSocket Comparison and Benchmarks.** [Online] Available: <http://blog.arungupta.me/rest-vs-websocket-comparison-benchmarks/>
- [17] [8] Hobbytronics. **Measuring Tilt Angle using Three Axis.** [Online] Available: <http://www.hobbytronics.co.uk/accelerometer-info>
- [10] [9] Bayesian Adventures. 2013. **Gyroscopes, Accelerometers and the Complementary Filter.** [Online] Available: <https://bayesianadventures.wordpress.com/2013/10/20/gyroscopes->

accelerometers-and-the-complementary-filter/

[10] GoomGum. 2016. **Kalman filter.** [Online] Available:

<https://goomgum.wordpress.com/2016/07/07/>



PRIMARY SOURCES

1	Submitted to Chulalongkorn University Student Paper	1 %
2	prezi.com Internet Source	<1 %
3	repository.rmutp.ac.th Internet Source	<1 %
4	Submitted to King Mongkut's Institute of Technology Ladkrabang Student Paper	<1 %
5	scitepress.org Internet Source	<1 %
6	intranet.dcy.go.th Internet Source	<1 %
7	kb.psu.ac.th Internet Source	<1 %
8	www.reportshop.co.kr Internet Source	<1 %
9	Submitted to Bangkok University	

<1 %

-
- 10 Submitted to Napier University <1 %
Student Paper
-
- 11 Submitted to University of Strathclyde <1 %
Student Paper
-
- 12 Koray Gürkan, Güray Gürkan, Ahmet Anıl Dindar. "Design and realization of multi-channel wireless data acquisition system for laboratory-scale experiments on structural health monitoring", Journal of Measurements in Engineering, 2018 <1 %
Publication
-
- 13 thaieasyelec.com <1 %
Internet Source
-
- 14 Submitted to University of Melbourne <1 %
Student Paper
-
- 15 www.emt.tugraz.at <1 %
Internet Source
-
- 16 cpe.rsu.ac.th <1 %
Internet Source
-
- 17 Submitted to Sheffield Hallam University <1 %
Student Paper
-
- 18 Submitted to Turun yliopisto <1 %
Student Paper

19	wiki.friendlyarm.com Internet Source	<1 %
20	www.mechatronics-lab.com Internet Source	<1 %
21	Submitted to Yonsei University Student Paper	<1 %
22	sveetch.net Internet Source	<1 %
23	Francesco Sorrentino, Louis M. Pecora, Aaron M. Hagerstrom, Thomas E. Murphy, Rajarshi Roy. "Complete characterization of the stability of cluster synchronization in complex dynamical networks", Science Advances, 2016 Publication	<1 %
24	cirrie.buffalo.edu Internet Source	<1 %
25	www.cs.utoronto.ca Internet Source	<1 %
26	ja.scribd.com Internet Source	<1 %
27	www.sl.ac.th Internet Source	<1 %
28	www.lawreform.go.th Internet Source	<1 %

29

www.phichit.go.th

Internet Source

<1 %

30

pajit.lpru.ac.th

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off